

Express Mailing Label No. EF243190205US

PATENT APPLICATION
Docket No. 1001.2.2

UNITED STATES PATENT APPLICATION

of

Kyle E. Powell

Mark B. Hurst

and

Karl G. Merkley

for

**INTERNET CONTENT DELIVERY ACCELERATION SYSTEM
EMPLOYING A HYBRID CONTENT SELECTION SCHEME**

Brian C. Kunzler, P.C.
ATTORNEY AT LAW
10 WEST 100 SOUTH, SUITE 425
SALT LAKE CITY, UTAH 84101

1 **INTERNET CONTENT DELIVERY ACCELERATION SYSTEM**
2 **EMPLOYING A HYBRID CONTENT SELECTION SCHEME**

3
4 **BACKGROUND OF THE INVENTION**

5
6 **1. The Field of the Invention**

7 The present invention relates to Internet acceleration systems and specifically, to
8 Internet acceleration systems involving satellite transmission and proxy caching to speed up
9 distribution of Internet content.
10

11 **2. The Relevant Art**

12 The Internet, or World-Wide Web as it is otherwise known, is one of the most
13 significant technological developments of modern times. It provides almost instantaneous
14 transfer of information across virtually the entire globe and enables direct communication
15 between people of different cities, countries, and even continents.

16 Nevertheless, the Internet is not without its problems. Due to the increasing use and
17 the basic point-to-point nature of the Internet, gridlock has hit. That is, too much Internet
18 traffic and too little bandwidth are substantially slowing down the response times for the
19 transfer of information over the Internet.

20 One solution that is being developed in response to this problem is caching. Caching
21 involves storing web pages and other frequently accessed digital content at or near the edge
22 of the Internet and close to users at businesses and ISP sites. Moving content to the edge of
23 the Internet in this manner dramatically reduces Internet traffic. With caching solutions,
24 Internet performance is improved through the use of local storage, rather than merely added
25 communication bandwidth. Such arrangements are efficient because the cost of storage is
26 becoming increasingly less expensive, while achieving greater bandwidth remains relatively
expensive.

1 While caching solutions appear promising for boosting Internet performance, two
2 major hurdles to the widespread use of caching currently exist. First, while certain entities
3 such as backbone operators that sell directly to large customers have sufficient "critical
4 mass" to benefit significantly from caching, others, such as businesses and smaller Internet
5 service providers (ISPs), do not. The success of web page caches is a function of "hit rate,"
6 the percentage of requests where the requested digital content is already present in cache.
7 But an enormous amount of digital content is available on the Web and caches servicing
8 smaller populations of varied end users tend to have much lower hit rates than caches
9 servicing large populations of end users.

10 The second hurdle is that in order to fill current caching systems, the digital content
11 of the cache is required to transit the Internet backbone at least once. Current caching
12 systems typically store the most recently requested digital objects, making place for those
13 objects by displacing the least recently used digital objects in the cache. Subsequent
14 references to objects that are stored in the cache in this manner are then able to avoid
15 traveling across the Internet and consequently have faster access times. Nevertheless, with
16 caches in small Internet communities and their attendant lower hit rates, requested objects
17 are frequently not present in the cache, and those objects must still transit the web getting to
18 the requesting user.

19 From the above discussion, it can be seen that new solutions for improving the speed
20 of caching on global information networks such as the Internet are needed. For instance, an
21 improved caching system that achieves increased hit rates of digital content within a cache
22 would be a great improvement in the art. An improved manner of extracting web objects and
23 distributing those objects to local caches would also be very helpful. Additionally, the ability
24 to transmit local popularity information to a central location where it can be compiled and
25 used to select web objects for transmission to local caches would also be helpful. It would
26 be particularly helpful to provide an improved manner of locally determining which digital
content is likely to be most popular to local users of a particular cache.

OBJECTS AND BRIEF SUMMARY OF THE INVENTION

The Internet content delivery acceleration system of the present invention has been developed in response to the present state of the art, and in particular, in response to the problems and needs in the art that have not yet been fully solved by currently available Internet content delivery acceleration systems. Accordingly, it is an overall object of the present invention to provide an Internet acceleration system and method that overcomes many or all of the above-discussed shortcomings in the art.

To achieve the foregoing object, and in accordance with the invention as embodied and broadly described herein in the preferred embodiment, an improved Internet content delivery acceleration system and method are provided. Within the system, a central proxy server, or caching system, selects high demand digital objects from the Internet and transmits those digital objects to a plurality of local proxy servers. The transmission of digital objects may be conducted using broadcast, multicast, reliable multicast, unicast, and reliable point to point transport over any suitable medium, including over the electromagnetic waves, fiber optics, and satellite transmission.

In one embodiment, the transmission utilizes a multicast protocol, such as IP multicast transmission from a geo-synchronous satellite. Thus, local content filling of the local proxy servers is provided by transmitting Internet objects to all subscribing local proxy servers at a high rate of speed. Through the use of multicast protocol, only subscribing or interested local proxy servers receive the transmission.

Preferably, the local proxy servers utilize a locally customizable priority determination or heuristic scheme to determine whether to keep or discard the digital objects. The priority determination preferably employs global popularity data, and may also utilize customizable local policies relevant to the particular customers subscribing to the local proxy server.

The local proxy servers are preferably provided with software that enables them to receive the digital objects at the high rate of speed and to store the digital objects in attendant

1 local cache databases for quickly servicing future digital object requests. In one
2 embodiment, the software comprises a cache database management module configured to
3 communicate directly with a cache database. The local proxy server software preferably also
4 comprises a local caching module in integral communication with the cache database
5 management module. Preferably, the integral communication comprises direct
6 communication between programs on a common server or other computer, and may be
7 facilitated by an applications program interface (API).

8 Because of the tight integration between the cache database management module and
9 the local caching module, sophisticated heuristics determinations may be employed. For
10 instance, rather than simply registering and reporting to the central proxy server when a
11 digital object is requested and is not present in a local cache, customized hit information for
12 all transmitted digital objects and even digital objects that have not been transmitted may
13 also be received from the local cache databases and transmitted to the central proxy server
14 for analysis and use in determining which digital objects to obtain and forward to the local
15 proxy servers. Additional determinations may be made regarding characteristics such as the
16 timing of demand for digital objects, together with the nature of the digital objects for
17 customized determinations of demand within the individual local proxy servers.

18 When a user requests Internet data, the request is first sent to the local cache
19 database management module to determine whether the requested digital objects is present
20 therein. If the digital objects is present, it is rapidly transmitted to the user directly from the
21 local cache database, without the necessity for transmission over the Internet. If the digital
22 object is not present, a request is transmitted to the central proxy server which requests a
23 copy from the hosting site.

24 The system may also transmit selected digital objects to subscribing local intranet
25 sites to rapidly and systematically publish private digital objects to local proxy servers
26 connected to the local intranets.

BRIEF DESCRIPTION OF THE DRAWINGS

In order that the manner in which the advantages and objects of the invention are obtained will be readily understood, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

Figure 1 is a schematic block diagram illustrating one embodiment of an Internet content delivery acceleration system of the present invention, including a central proxy server and a plurality of remote proxy servers.

Figure 2 is a schematic block diagram illustrating a further embodiment of an Internet content delivery acceleration system of the present invention, including private intranet facilities.

Figure 3 is a schematic block diagram illustrating functional components of one embodiment of software used with the systems of Figures 1 and 2.

Figure 4 is a schematic block diagram illustrating a packet for satellite pointcast to local private intranets.

Figure 5 is a schematic block diagram illustrating the functional components of one embodiment of a priority determination procedure of the present invention.

Figure 6 is a schematic flow chart diagram illustrating the operation of one embodiment of software for use on the local proxy server of Figure 1.

Figure 7 is a schematic flow chart diagram illustrating the operation of one embodiment of software for use with the central proxy server of Figure 1.

Figure 8 is a schematic block diagram illustrating one embodiment of a central proxy server of the present invention.

1 Figure 9 is an illustration of the structure of one embodiment of a communications
2 packet suitable for use in conjunction with the present invention.

3 Figure 10 is a schematic block diagram of one embodiment of a local proxy server
4 of the present invention.
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26

Under a preferred embodiment of the present invention, the communication medium comprises transmission by a satellite 18 at a high rate of speed enabled by efficient hard drive data receipt and storage methods encompassed within the present invention. In one

embodiment, this rate of speed is about 25 Mbps. A preferred manner of transmission is IP multicast.

The web pages 15a are preferably concurrently transmitted to each subscribing local proxy server 22 once any of the local proxy servers 22 requests the web pages 15a. Each local proxy server 22 receives the web pages 15a and stores them in its local cache database 24 for a selected amount of time before considering whether to purge the particular web pages 15a. In one embodiment, the selected amount of time is scaled to the amount of memory residing in the particular local cache database 22.

The local proxy servers 22 in one embodiment utilize proxy caching protocol built into the HTTP Internet language for proxy servers. Under this protocol, every time a user at a station 30 requests a web page 15a over the network of which the local proxy server 22 is a part, the request passes through the local proxy server 22. As part of the protocol, the local proxy server 22 initially checks its attendant local cache database 24 to determine whether the web page 15a is present. In one embodiment, this consists of the local caching module 17 checking through the integral communication interface with the cache database management program 29.

If the web page 15a is present, the local proxy server 22 immediately transmits the web page 15a through the network 44 to the user at a station 30. Thus, the request is transparent and to the user at a station 30, it appears as if the web page 15a was transmitted over the Internet 20. Advantageously, by bypassing the Internet 20, the transmission is much more rapid. Additionally, Internet traffic is reduced, and Internet connection costs are potentially much lower.

If the requested web page 15a is not present in the local cache database 24, the HTTP caching protocol causes the request to be passed on through the Internet 20 to the Internet site 35 at which the web page 15a is hosted. The Internet site 35 then transmits the web page 15a over the Internet 20 back to the requesting user at a station 30. This Internet transmission occurs over regular Internet communication channels 38, 40, 42.

1 Concurrently, the request is also passed to the central proxy server 12. Once the
2 central proxy server 12 receives its copy of the request, it also requests a copy of the
3 requested web page 15a from the hosting Internet site 35. Accordingly, the web page 15a is
4 also transmitted over channels 36, 37 to the central proxy server 12. Once the web page 15a
5 is received by the control proxy server 12, the central proxy server 12 caches the web page
6 15a within the attendant master cache database 14. The central proxy server 12 may also
7 multicast the web page 15a to the communicating local proxy servers through the
8 communication medium if the web page 15a is found to have a sufficiently high priority.
9 The communication medium in one embodiment is satellite transmission. The web page 15a
10 is transmitted 32 through a satellite uplink transmitter 16 to the satellite 18 at a speed of, for
11 example, 25 Mega bytes per second (Mbps). The satellite 18 then transmits 34 the web page
12 15a to each of the subscribing local proxy servers 22, at a high rate such as 25 Mbps.

13 The central proxy server 12 may filter the web page 15a before caching it. It may
14 also keep track of the number of requests for the web page 15a and transmit web pages 15a
15 (or other digital objects 15) only after a certain minimum number of requests have been
16 logged for that web page 15a. Accordingly, the central proxy server 12 is in a position to be
17 a ratings system for the Internet 20, having centralized access to web site demand
18 information.

19 Additionally, the central proxy server 12 preferably receives global popularity
20 information about digital objects 15 from subscribing local proxy servers 22. This popularity
21 information in a basic embodiment comprises miss data. Additionally, due to the unique
22 configuration of the present invention, more sophisticated information may also be
23 transmitted. This may include, for instance, hit information and timing information, as will
24 be discussed in greater detail below.

25 The central proxy server may also utilize digital object popularity information,
26 digital object characteristics, data associations, and other data useful to local proxy servers
22 in deciding which extracted digital objects 15 to continue to store. Such information may

1 be collected from the local proxy servers 22, and in addition, may also be received from
2 companion digital object extractor servers 35 communicating with the central proxy server
3 12.

4 The digital object extractor servers 35 preferably locate information relevant to the
5 priority determinations of the local proxy servers 22 and pass that information to the central
6 proxy server for compilation and later transmission to the local proxy servers 22 and
7 optionally, to other requesting entities, possibly for a subscription fee. Manners in which this
8 information may be gleaned include directly contacting web sites that contain hit rates or
9 other popularity information. Additionally, the digital object extractor servers 35 may
10 themselves traverse or "crawl" over the web to examine web sites and observe traffic. When
11 examining web sites, the digital object extractor servers 35 may follow links on those web
12 sites to other web sites to similarly examine the other web sites. This process may be
13 continuous, and the information that is gathered is preferably passed to the central proxy
14 server, as stated.

15 Additional information that may be collected by the central proxy server 12 includes
16 the qualitative information regarding the digital objects 15, rather than just quantitative data.
17 Thus, types of digital objects 15 may be collected, either from the local proxy servers 22 or
18 from the digital object extractor servers 35. This data may include, for instance, the subject
19 matter of the digital objects 15, the types of sites requesting the digital objects 15, etc. This
20 information may then be used by the local proxy servers 22 in making a customized priority
21 determination according to local demand and local user types.

22 In the depicted embodiment, a single central proxy server services the entire system
23 10. Nevertheless, more than one central proxy server 12 may be in operation. For example,
24 different geographical locations may be served by different central proxy servers 12. These
25 multiple central proxy servers 12 may be in communication with each other either through
26 suitable mediums and may share information such as hit and miss rate information and other
useful content selection information.

1 Each local proxy server 22 is preferably provided with a communication facility,
2 such as a satellite down-link receiver 26. In the depicted embodiment, the down-link
3 receiver 26 receives the satellite multicast transmission 34 of the extracted web pages 15a.
4 Upon receiving the web page 15a, each subscribing local proxy server 22 caches the web
5 page 15a for a selected period of time for access by users at the communicating stations 30.
6 At the end of the selected period of time, each local proxy server 22 preferably utilizes an
7 individualized local policy and a priority determination program to determine whether to
8 keep or discard the transmitted web page 15a.

9 Reference is now made to Figure 2 for a discussion of a further aspect of the present
10 invention. In addition to multicasting generally to a series of local proxy servers 22, the
11 central proxy server 12 may also service one or more virtual private intranets 50 through
12 selective pointcast satellite transmission 74. Under this concept, and in an embodiment to
13 be specifically described herein, the system 10 preferably operates in substantially the same
14 manner as described above. Accordingly, the system 10 is provided with the central proxy
15 server 12 and a series of local proxy servers 22 configured substantially in the manner
16 described. In addition, the system 10 may also be provided with the private intranet 50. The
17 private intranet 50 is shown communicating over the Internet 20 through an intranet proxy
18 server 60. The intranet proxy server 60 may also function as a local proxy server 22, as
19 discussed above.

20 As depicted, the private intranet 50 connects to end users at remote stations 52
21 through a communication channel 54. The private intranet 50 is connected with the intranet
22 proxy server 60 through a communication channel 68. The Intranet proxy server is in
23 communication with a satellite down-link receiver 62 through a communication channel 66
24 and with the Internet 20 with a communications channel 64. One or more of the
25 aforementioned channels may maintain security with a firewall 78.

26 The system 10 may also be provided with a private intranet publisher 70 for
generating or otherwise providing private digital objects 55 to be communicated to users at



1 stations 52 within the private intranet 50. In the depicted embodiment, the private intranet
2 publisher 70 is a dedicated server computer communicating over the Internet 20 through a
3 secure channel 72. The secure channel 72 is shown provided with a firewall 78.

4 The private intranet publisher 50 generates or collects the private digital objects 55,
5 which may be business, financial, or any other type of data, and transmits the private digital
6 objects 55 in web page form. The private digital objects 55 is passed in a secure manner
7 through the Internet 20 to the central proxy server 12. The central proxy server 12, through
8 a satellite uplink transmitter 16, then uplinks 32 the private digital objects 55 to the satellite
9 18. The satellite 18 then transmits the private digital objects 55 to intranet proxy servers 60
10 of the subscribing virtual private intranet using a focused multicast 74. The same satellite
11 18 may also transmit 34 other digital objects such as web pages 15a to the regular local proxy
12 servers 22.

13 All private intranet transmission over the Internet 20 is preferably kept within a
14 firewall 78, preferably through hashing or other encryption of the digital objects being
15 transmitted. The multicast transmission 74 could be transmission by a private frequency, or
16 more preferably, through the same frequency and channels as the normal web site multicasts
17 34, but with a header placed on each transmitted packet identifying the transmission as
18 private and identifying the designee. In such a case, encryption may be employed, and the
19 remote intranets intended to receive the digital objects may be provided with encryption keys.
20 A master key is preferably retained by the central proxy server 12. Local proxy servers 22
21 for which the private digital objects 55 is not intended thus ignore the digital objects 55. The
22 header could also particularly specify whether the digital objects is private digital objects 55
23 or global digital objects 15.

24 Thus, the Virtual Private Intranet of the present invention, unlike prior art broadcast
25 intranets, is one-way, and the private digital objects 55 to be transmitted are collected or
26 otherwise designated by the central publisher 70. In one embodiment, the publisher 70

1 collects private digital objects 55 as a result of requests from remote intranet sites, possibly
2 over the Internet 20.

3 One example of the use of a virtual private intranet 50 of the present invention is
4 within a car dealership chain. In such a case, private digital objects 55 such as, for example,
5 sales and pricing data may be transmitted over the virtual private intranet 50. Financial
6 advising companies may also use the system and pointcast 74 digital objects 55 in the form
7 of market data or financial analysis in another example. Banks may make financial
8 transactions with the system 10. As a further example, businesses might transmit sales
9 information, company policies, advertising materials, etc., over the virtual private intranet
10 50.

11 It is preferred that the private digital objects 55 are pointcast 74 transparently to the
12 users 52. Thus, the digital objects 15 may be viewed using the file transfer protocol of the
13 local network, rather than as an Internet URL site. Accordingly, users 52, though possibly
14 located across the world from each other, may access the private digital objects 55 from the
15 proxy server as if the objects 55 are a local part of the remote intranet 50. The private digital
16 objects 55 may be transmitted on demand, but it is contemplated that at least a substantial
17 portion of the private digital objects 55 may be transmitted through the discretion of the
18 Internet publisher 70, or may comprise standard information, periodically updated and
19 transmitted.

20 Referring now to Figure 3, shown therein is a schematic block diagram illustrating
21 more specifically the various functional components of one embodiment of software used to
22 enable the system 10 of the present invention. The software modules contained in the
23 schematic block diagram of Figure 3 are generally implemented as software instructions,
24 procedures, routines, or other executable software code. Nevertheless, the modules could
25 also be implemented with other types of programmable logic such as programmable logic
26 arrays (PLAs), ASICs, or even logic circuits or other types of hardware.

1 In order to initialize the system 10 (of Figures 1 and 2) and enable communication
2 links, the depicted components of a system initialization block 80 is employed. The modules
3 of the system initialization block 80 may be contained as software code within the central
4 proxy server 12, or may be distributed amongst the different hardware components of the
5 system 10. Initially, the central proxy server 12 is initialized and brought on-line with a
6 central server enablement module 82. This preferably includes enabling Internet
7 communication over communication channel 36 and enabling satellite communication with
8 the satellite uplink transmitter 16.

9 Subsequently, the local proxy servers 22 are enabled and brought on-line with a local
10 proxy server enablement module 84. Typically, Internet communications are enabled
11 through communication channels 38, 40, and 42. It is of note that the various local proxy
12 servers 22 will generally not all be brought on-line at one time. Typically, the system 10 of
13 the present invention will be provided commercially as a service to which customers
14 subscribe. As new customers subscribe, they are provided with the local proxy server 22 and
15 satellite receiver hardware 26 for an initial fee. A monthly subscription fee may be charged
16 thereafter.

17 Satellite communications are enabled with a satellite initialization module 86. This
18 may include establishing the communications link between the satellite 18 and the satellite
19 uplink transmitter 16 and between the satellite 18 and the satellite receivers 26, 62. It may
20 also include establishing a proper protocol for satellite communications. For the embodiment
21 of Figure 2, satellite communications for the pointcast transmission must be established with
22 one or more Internet proxy servers 60.

23 The intent of the system 10 is to fill the local cache databases 24 of each subscribing
24 local proxy server 22 with the most highly requested digital objects such that the majority of
25 requests for web pages 15a may be serviced directly by the local proxy servers without having
26 to resort to the Internet 20. Accordingly, each new central proxy server that is brought on-
line must be initially filled with high demand web content 15a. An initial filling module 88

1 and may be utilized for this purpose and may function in a variety of manners. In one
2 contemplated embodiment, the central proxy server 12 transmits “old” digital objects that
3 have been previously transmitted in between transmitting “fresh” digital objects that have
4 only recently been requested. This transmission may be conducted in bursts to transmit the
5 entire contents of the master cache database systematically, or according to a selected
6 heuristic formula.

7 A user station block 90 of Figure 3 illustrates the basic functional components and
8 operation of a user station 30 of Figure 1. Through an Internet digital object request module
9 92, the user at the end station 30 places a request for a digital object 15. In one embodiment,
10 the digital object 15 is a web page 15a and the digital object request module comprises a
11 standard web browser 25 which is connected to the Internet through a local proxy server 22.
12 The digital object 15 is transparently received from the local proxy server 22 through a digital
13 object reception module 94. The digital object reception module 94 may comprise the web
14 browser 25 in conjunction with the HTTP proxy caching protocol which searches local cache
15 databases 24 prior to transmitting a request for Internet data over the Internet 20.

16 Once the web page 15a is received, either from the local cache database 24 or over
17 the Internet 20, it is displayed to the user with a digital object display module 96. The digital
18 object display module 96 in one embodiment comprises the web browser 25 together with
19 the proxy caching protocol of the HTTP language operating on a PC or other computer.

20 A central server block 100 illustrates the basic functional components operating
21 within the central proxy server 12 of Figure 1. These components are preferably part of a
22 central caching module 13 of Figure 1. A digital object request reception module 102 allows
23 the central proxy server 12 to receive the request for a digital object made by the user at a
24 station 30 with the digital object request module 92. An Internet request module 104 passes
25 the request on through the Internet 20. A digital object reception module 106 receives the
26 digital object 15 transmitted by the Internet 20 in response. A cache storage module 108
receives the requested digital object 15 and stores a copy of the digital object 15 within the



1 master cache database 14. An uplink transmission control module 109 coordinates with the
2 communication network to distribute selected data 15 to the local proxy servers 22. In one
3 embodiment, the uplink transmission control module 109 communicates with the satellite
4 transmitter 16 in transmitting digital objects 15 through the uplink 32 to the satellite 18. Of
5 course, other types of transmission could be utilized, including transmission over the Internet
6 itself.

7 A satellite operation block 110 illustrates the basic operation of functional modules
8 operating within the satellite 18 of Figures 1 and 2. Once the web page 15a has been
9 uplinked by the central proxy server, the satellite 18, an uplink reception module 112 receives
10 the uplinked web page 15a or other digital object 15. Subsequently, a digital object
11 transmission module 114 formats and transmits the digital object 15 through satellite
12 multicast 34 to all associated local proxy servers 22. Additionally, if the uplinked digital
13 object 15 contains private digital objects 55, a pointcast data module 116 formats and
14 pointcasts the private digital objects 55.

15 An uplink transmitter block 120 of Figure 3 illustrates general functional modules
16 operating within the satellite uplink transmitter 16 of Figures 1 and 2. The digital object 15
17 to be uplinked is formatted into a proper form and protocol for satellite transmission with a
18 digital object formatting module 124. Header information is added to the packets to be
19 uplinked with a header addition module 124. One simplified example of the structure of a
20 packet 190 is shown in Figure 4. A typical packet 190 is comprised of data 191 preceded by
21 a header 192. The uplink transmission is conducted between the satellite uplink transmitter
22 and the satellite 18 with the use of an uplink transmission module 126.

23 An Internet block 130 illustrates one example of the functional modules operating
24 within the Internet 20 of Figures 1 and 2. Requests for digital object 15 are transmitted over
25 the Internet 20 with a digital object request module 132. The request is typically the request
26 generated by the Internet request module 104. Once the digital object 15 is requested, the
Internet site 35 at which the digital object 15 is hosted is contacted with a site contacting

1 module 134. A site map of the site, including the location and configuration of the web page
2 15a is transmitted from the web site 35 with a site map transmission module 136. Thereafter,
3 the web page 15a is transmitted from the Internet site 35 to the central proxy server 12 with
4 a digital object packet transmission module 138. The web page 15a and attendant site map
5 may also be transmitted directly to the requesting local proxy server 22.

6 A local server block 150 illustrates one embodiment of the functional components
7 operating within the local proxy server 22 of Figures 1 and 2. Preferably, these functional
8 components are contained within the local caching module 17 of Figure 1. Within the local
9 server block 150, a digital object request block 155 includes a digital object reception module
10 152. The digital object reception module 152 receives requests for digital object 15 from the
11 end users at the stations 30. Generally, this request is generated by the digital object request
12 module 92. A search module 154 searches the local cache database 24 for the requested
13 digital object 15. In one embodiment, the search module 154 comprises the proxy cache
14 protocol employed within the HTTP language.

15 If the requested digital object 15 is not present within the local cache database 24,
16 a digital object request module 156 transmits a request for the digital object 15 to the central
17 proxy server 12. This request is typically routed over one of the communication channels 38,
18 40, or 42, through the Internet 20, and through communication channel 36 to the central
19 proxy server 11. The request may comprise the original request received from the user at a
20 station 30 modified with the Internet URL of the central proxy server 12. Concurrently, the
21 digital object request module 156 may also request the digital object 15 directly from the
22 Internet site 35 where the digital object 15 is hosted.

23 A digital object reception module 158 receives the requested digital object 15.
24 Typically, the digital object 15 is provided by the central proxy server 12 in accordance with
25 the procedure discussed for the central server module 100. Thus, the digital object 15 is
26 uplinked 32 to the satellite 18, which multicasts the digital object 15 to all subscribing local
proxy servers 22. The digital object 15 is received by the satellite receiver 26 and passed to

1 the local proxy server 22. The local proxy server 22 may also receive the digital object 15
2 directly over the Internet. The digital object 15 from the earliest arriving of the multicast 18
3 and the direct Internet transmission is then passed on to the user at the station 30.

4 A digital object management block 160 illustrates the functional components of the
5 local proxy servers 22 which handle the digital object 15 received from the satellite
6 transmission 34. When a digital object 15 is requested by a user station 30 and is found to
7 not be present within the attendant local cache database 24, a request is made for the digital
8 object to the central proxy server 12, as discussed. That digital object 15 is then requested
9 from the Internet site 35 where it is hosted, as also discussed, and then transmitted through
10 satellite transmission 34 to the local proxy server 22 requesting the data. Concurrently, the
11 digital object is also received by the satellite receivers 26 of all other subscribing local proxy
12 servers 22 with the use of the digital object multicast reception module 162 which is
13 preferably programmed into each of the local proxy servers 22. Typically, all of the satellite
14 receivers 26 are attuned to the same frequency over which the digital object 15 is transmitted.
15 Of course, if a number of satellites 18 are used to transmit the digital object 15 around the
16 world, a number of different frequencies may also be employed.

17 Once the digital object 15 is received, a digital object supervisor module 164
18 operating within each of the local proxy servers 22 attends to the storage of the digital object
19 15 within the local cache database 24. The digital object 15 is automatically stored for a
20 predetermined amount of time. In one example, the predetermined amount of time may be
21 is a period of four hours. Once that time has passed, a priority determination application
22 module 166 conducts a priority determination to determine whether to continue to store or
23 to discard the digital object 15. If, after application of the priority determination, it is decided
24 that the likelihood of a request for the digital object 15 is low for that particular local proxy
25 server 22, a push module 168 discards the digital object 15, removing it from the local cache
26 database 24. In one preferred embodiment, each local proxy server 22 is provided with its
own, individualized priority determination scheme.

1 In one embodiment, the modules 158, 160 integrally communicate with a cache
2 database management module such as Novell's ICS™ software, which in turn attends to the
3 storage of the digital object 15 onto disk at the high rate of speeds specified, and/or Novell's
4 Border Manager™ software which acts as a proxy cache manager and provides proxy content
5 filling and firewall functions.

6 A local relevance determination block 170 illustrates the functional components of
7 one embodiment of a priority determination scheme of the present invention. A local
8 statistics collection module 172 resident within each local proxy server 22 monitors the
9 frequency of requests for each file of digital object 15 located within the local cache database
10 24. A global statistics reception module 174 receives digital object demand statistics from
11 a central location. In one embodiment, the central location is the central proxy server 12,
12 which collects demand statistics during the process of servicing requests for digital objects
13 15. In this manner, the central proxy server 12 acts as a sort of "Nielsen Ratings Service" for
14 the Internet. This information may be provided either free or for a fee to interested parties as
15 will be discussed in greater detail below.

16 A priority determination module 176 utilizes the statistical information gathered by
17 modules 172 and 174 in making the priority determination of whether to keep a particular
18 digital object 15. If the determination is to keep the digital object 15, a least recently used
19 (LRU) module may be used to decide which existing digital object is to be discarded to make
20 room for the new digital object 15. The LRU calculation module 178 calculates the amount
21 of use of each file of digital object 15 and adds a local use factor into the determination. This
22 priority determination is applied by the priority scheme application module 166 as described
23 above.

24 The priority determination module 176 preferably utilizes localized web page
25 demand statistics compiled by the local proxy server 22 as well as globalized web page
26 demand statistics collected by and transmitted periodically from the central proxy server
through satellite transmission.

1 A hit rate reporting module 179 periodically reports the local hit rates for part or all
2 of the digital objects 15 transmitted from the central proxy server 12. This information is
3 used to calculate global demand statistics. Preferably, each participating local proxy server
4 22 tabulates every request from every end user station 30. A hit report is then periodically
5 transmitted to the central proxy server 12. The central proxy server 12 in turn tabulates the
6 accumulated hit reports for all web pages requested and periodically sends updated usage
7 information with the hit totals to the local proxy servers 22. Preferably, every digital object
8 15 is assigned a globally unique identification number which can be stored and transmitted
9 compactly and which is used to identify the particular digital object 15.

10 It is preferred that the relevance determination module 170 is closely integrated with
11 the digital object management module 160. In one embodiment, the digital object
12 management module 160 is implemented through tight integration with the cache database
13 management module program 29 and the priority determination module is configured to
14 work closely with the cache database management module 29.

15 In one embodiment, the digital object management module 160 keeps track of related
16 files within a web page 15a or related to a web page 15a. The related files may be uplinked
17 and transmitted together through satellite transmission 34. The relevance determination
18 module 170, being integrated with the digital object management module 160, may be
19 configured to receive a list of the objects and files associated with a page. In this manner,
20 the local digital object relevance determination module 170 may be able to ensure that the
21 related digital objects stay together and are stored together in the local cache database 24.
22 It may also calculate usage information for each of the objects and files separately and keep
23 all associated files or “trim” a portion of the objects and files that are not highly used. The
24 related files may then be transmitted as a group when one or more of the related files is
25 requested by a user at a communicating station 30.

26 Additionally, the local relevance determination module 170 may also track
“supergrouping” of web page information 15. In many web pages, information such as

1 advertisements change or alternate back and forth. These changes may be kept track of and
2 the differing versions transmitted together by the central proxy server 12.

3 One example of a manner of configuring the relevance determination module 170 is
4 illustrated in Figure 5. Shown therein is relevance determination module 170 containing a
5 local demand data module 202. In addition, each local proxy server 22 may also have a
6 custom local policy 204. Within the policy 204 is a relative weighting 206 of local and
7 global web page demand. This policy 204, the local demand data 202, as well as global
8 demand data 208 are preferably employed by the priority determination module 200 to
9 determine whether to keep or discard each separate digital object 15 that is received, once
10 the selected period of time has passed. The policy 204 may also weight various subject
11 matters attributed to the web pages 15a. The policy 204 may also include other local value
12 factors, such as the ease of getting the digital object 15 for the particular local proxy server
13 22.

14 The policy 204 may also specify the particular interest areas that the priority
15 determination scheme is intended to weight. For instance, in a case where the local proxy
16 server 22 services a school, the policy 204 may give greater weight to digital objects 15
17 containing educational issues. As a further example, if the local proxy server 22 services a
18 research institution, the policy 204 may give greater weight to digital objects 15 relating to
19 the particular research being conducted.

20 Thus, as shown in a global demand calculation block 180 of Figure 3, a global
21 demand monitoring module 182 resident within the central proxy server 12 monitors the
22 requests for information from each local proxy server 22. It also monitors the hit rate data
23 transmitted by the reporting modules 179 of each local proxy server 22. In response, a global
24 demand compilation module compiles the data collected by the global demand monitoring
25 module 182. Within those statistics may be information relevant to each particular subject
26 matter such that the priority determination schemes may take the categorized demand data

1 into account according to the particular weighting of the subject categories in their local
2 policies 204.

3 A particular subscribing local proxy server 22 might be an educational institution,
4 business, or news provider, and might weight subject matter statistics from relevant
5 categories more heavily within its own local policy 204 than others of the local proxy servers
6 22. A global demand transmission module 186 transmits the demand data, preferably by
7 satellite transmission 34, in the same manner that the requested digital object 15 is
8 transmitted.

9 Figure 6 is a schematic flow chart diagram illustrating one manner of operation of a
10 software program 220, a copy of which preferably operates on each subscribing local proxy
11 server 22. In one embodiment, the software program 220 corresponds to the local caching
12 module 17. The program 220 initializes at a start block 222 then proceeds to a block 224,
13 where it awaits receipt of input to be processed. Input may be received in the form of a
14 terminate command, which is received at a block 226. The program 220 then progresses to
15 an end block 228, where it terminates.

16 Several additional functions may also be accomplished by the program 220. For
17 instance, in one embodiment, the program 220 may receive a user request for a digital object
18 15, as depicted at a block 230. The program 220 then proceeds to a block 232, where the
19 caching protocol of the HTTP language is utilized to consult the local cache database 24 and
20 thereby determine if the requested digital object 15 is present. At a query block 234, the
21 program 220 branches in one of two directions, depending on whether or not the digital
22 object 15 is present in the local cache database 24. If the digital object 15 is present, the
23 program 220 proceeds to a block 236 where it transmits the digital object 15 to the requesting
24 station 30 for presentation to the user. The program then proceeds to a block 238, where
25 control of the program is returned to the start block 222.

26 If the digital object 15 is not present, the program 220 proceeds from the query block
234 to a block 240, where it passes the request on through the Internet 20 to the hosting site

1 35. At about the same time, the program 220, at a block 242, transmits a copy of the request
2 to the central proxy server 12. The program 220 then waits at a block 244 for the requested
3 digital object 15 to return by way of the fastest route, whether it be through satellite
4 transmission 34 or through the Internet 20. Once the digital object 15 is received from the
5 fastest route, the program 220 transmits the digital object 15 to the requesting user station
6 30 for presentation to the user. Thereafter, the program 220 moves to a block 247 where
7 local demand statistics are updated.

8 At a step 248, the local proxy server 22 transmits hit information to the central proxy
9 server 12. Miss information, resulting from a request for which nor corresponding object 15
10 is present in the local proxy cache 24, is communicated to the central proxy server 12 by the
11 request of step 242. In addition, the local proxy server 22, preferably through the integral
12 interface with the cache management module 29, may be configured to transmit miss
13 information to the central proxy server 12 for compilation into global demand data. In one
14 embodiment, hits are compiled into a file which is of a certain size. When the file is full, or
15 at selected intervals, the miss report is transported across the Internet 20 to the central proxy
16 server 12. All local proxy caches 22 in one embodiment continually send such reports, acting
17 as the eyes and ears of the system 10. The local caching module 17 may also compile reports
18 of hits and misses for use in determining local relevance of objects transmitted from the
19 central proxy server 12. Subsequently, the program moves to a block 249 where control is
20 passed back to the receive input block 224.

21 The program 220 may also receive as input the transmission of web pages 15a or
22 other digital objects 15 from the central proxy server. In one embodiment, the digital object
23 15 is received at a block 250. The program 220 then proceeds to a block 252 where it stores
24 the received digital object 15 in the local cache database 24 for a selected amount of time.
25 After the selected amount of time has passed, the program 220 generates feedback data
26 utilizing the priority determination scheme 200 of Figure 5, and makes a determination of
whether to retain or discard the digital object 15. Preferably, this priority determination is

1 localized to the particular local proxy server 22, and may take the form of a local relevance
2 determination. Of course, the priority determination could be made prior to storing the
3 digital object 15, and if the object is of insufficient interest, the digital object 15 is discarded
4 rather than stored. One example of a local relevance determination is illustrated in Figure
5 12 and is described below.

6 Once the priority/relevance determination is made, the digital object 15 is either
7 stored for a greater period of time or is discarded at a block 256. If the priority determination
8 results in the decision to store the digital object 15 and the local cache database 24 is
9 currently fully filled with digital objects 15, a LRU procedure (e.g., module 178 of Figure 3)
10 may be used to discard the least recently used digital object 15 within the database 24 or the
11 digital object for which the LRU procedure otherwise determines is the least likely to be
12 requested in the future. Thereafter, the program 220 progresses to a block 258 which returns
13 control to the start block 222. In one embodiment, the determination of which objects to
14 discard is made by the caching program, which may be modified to take into account global
15 demand statistics, as well as local usage, and a threshold value statically or dynamically set
16 by the priority determination module 176.

17 If the input received by the receive input block 225 is global demand statistics from
18 the central proxy server 12, the program 220 progresses to a block 260, where the global
19 demand statistics are received. Subsequently, at a block 262, the local record of global
20 demand statistics is updated. At a block 264, the local use statistics generated at block 247
21 are consulted and obtained. The local use statistics are compiled together with the global
22 demand statistics at a block 266. The priority determination procedure 200 of Figure 5 is
23 then employed at a block 268 to calculate the local hit potential of the particular digital object
24 15. The hit potential is then compiled as priority data for use in block 254 (and module 166
25 of Figure 3). The priority data is then stored and copies are passed on to the modules which
26 must employ the priority data to make a priority determination at a block 272. The program
220 then proceeds to a block 274 which passes control back to the receive input block 224.

1 Figure 7 is a flow chart diagram illustrating one manner of operation of a software
2 program 300 resident within and operating on the central proxy server 12 of Figures 1 and
3 2. The program 300 initializes at a start block 302 and proceeds to a block 304, where it
4 awaits receipt of a request for a digital object 15. Of course, the program 300 may have other
5 means of identifying high demand digital objects 15. Web site search engines or other pre-
6 generated statistical data may be consulted, for instance. The central proxy server 12 may
7 also keep a history of requests for web pages 15a, which are updated daily or weekly, etc. in
8 order to re-transmit those pages 15a once updated.

9 Once a digital object 15 is requested by a local proxy server 22 or otherwise identified
10 as being of sufficient demand to merit caching within the subscribing local proxy servers'
11 attendant cache databases 24, the program 300 proceeds to a block 305. Here, the program
12 300 requests a copy of the digital object 15 from the hosting Internet site 35. Subsequently,
13 as designated by a block 306, the digital object 15 is received over the Internet 20 from the
14 hosting site 35. The digital object 15 may then be filtered before being transmitted and/or
15 stored within the master cache database 312. For instance, a morality filter could be used to
16 filter out obscene language, hate content, pornographic materials, and the like. Other types
17 of filters may also be employed to filter by content, or filters could be used to filter out digital
18 objects for which it is known that hit rates in the future will be low. A heuristic scheme or
19 a request history might be employed for so doing.

20 At a block 310, the digital object 15 which has successfully passed through filtering
21 is stored in the master cache database 14 as represented at a block 210. The digital object
22 15 is then sent to the uplink transmitter 16 as represented by a block 312. As discussed, the
23 communication medium may be any suitable medium, but satellite transmission will be
24 discussed herein by way of example. As shown by a block 314, the digital object 15 is then
25 transmitted to the satellite 18. Thereafter, as shown by a block 316, the digital object 15 is
26 transmitted by satellite transmission 34 to the subscribing local proxy servers 22. This
transmission may be coordinated by the program 300.

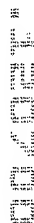
1 At a subsequent block 318, global demand statistics are updated to reflect the request
2 for the digital object 15. The updated global demand statistics may be transmitted at this
3 point to the local proxy servers, or this information may be periodically transmitted.
4 Subsequently, the program 300 progresses to a block 322 which returns control to the start
5 block 302.

6 The system 10 of the present invention moves high demand web pages to the edge
7 of the Internet, closer to users. The result is a much faster delivery of a majority of web
8 pages requested by end users at a station 30, and a corresponding decrease in Internet traffic,
9 substantially accelerating the Internet. Connection costs will correspondingly decrease,
10 resulting in savings to users employing the system 10.

11 Example of Central Proxy Server Operation

12 Referring to Figure 8, shown therein is one representative example of a manner of
13 operation of the central proxy server 12 of Figure 1. Within Figure 8 is shown a number of
14 local proxy servers 22 communicating with a central proxy server 12 via a
15 communications/validation manager 332. The communications/validation manager 332
16 maintains a secure connection from the central proxy server 12 to each of the local proxy
17 servers 22 in the system. Messages from the local proxy servers 22 are communicated
18 through the message switcher 338 to the appropriate module.

19 One of the primary modules for handling messages is the miss/refresh handler 340.
20 When a local proxy server 22 is missing any requested digital object 15, a miss is then passed
21 on to the central proxy server 12 and received by the miss handler 340. The miss handler 340
22 then goes out to the cache database management module 29, represented herein as an Internet
23 Caching System (ICS), one example of which is the ICS program available from Novell
24 Corporation of Provo, Utah, and requests the digital object associated with that miss. The
25 cache database management module of the central proxy server will either have that digital
26



1 object 15 locally in the local cache database 22 or it will go out to the Internet 20 and request
2 that digital object 15.

3 When that digital object is received, all components needed to assemble that
4 particular digital object 15 are also preferably received and assembled. The complete set of
5 components of the digital object 15 is then stored in the database 334, and a representation
6 of the object 15 (e.g., an object with the priority of the object as its key using C++ object
7 oriented programming, in one embodiment) is passed on to a priority queue 342. The priority
8 queue 342 handles all of the prioritization of various HTML pages 15a or other digital
9 objects 15 for later transmission.

10 When a particular digital object 15 has a sufficiently high priority, it is placed into
11 the transmission queue 342, which can then request all of the digital objects 15 associated
12 with this HTML page 15a, queue them up for transmission, and send them out to the
13 packetizer 348, where they are placed into full transmission packets and sent off to the
14 communication medium (in one embodiment, the satellite 18). Preferably, hits for the
15 objects 15 are still received while the object is in the transmission queue, 346, and the
16 object's priority can still be implemented. When the object 15 is transmitted, the object is
17 preferably placed back in the transmission queue 342 with an adjusted priority reflecting the
18 fact that the object 15 has been recently transmitted. In one embodiment, the priority queue
19 takes the form of a binary tree, as will be described below.

20 The central proxy server 12 preferably also includes a control sub-system 350 which
21 handles all of the start-up, shutdown, initialization and processing. An attention sub-system
22 352 is also preferably provided and handles all user interface types of interaction. The
23 central proxy server 12 is preferably in communication with a database 334 which may
24 employ a schema for handling historical logging of information transmitted back from the
25 local proxy servers 22 to the database 334 upon a message that is sent by the central proxy
26 server 12.

1 The digital objects 15 which may be handled and forwarded through the system
2 include web objects or groups of web objects that form a web page 15a. Web objects are also
3 referred to as HTML objects or HTML data.

4 When a digital object 15 is placed on the priority queue, the priority of that digital
5 object 15 is altered somewhat continuously as the central proxy server 12 receives indication
6 from the local proxy servers 22 that one or more end users have made a cache hit on that
7 object or that a local proxy server 22 reported another miss. This popularity information
8 keeps accumulating even after the initial insertion into the priority queue and the highest
9 priority digital objects 15 are considered for transmission.

10 The transmission queue 346 draws the highest priority digital objects 15 off the
11 priority queue and then attempts to gather together the web objects, container objects, and
12 their component objects and form that digital object 15 into logical messages. The input side
13 of the transmission queue 346 receives the digital object 15. The output of the transmission
14 queue 346 is a list or multiple lists of logical messages that represent pieces of that digital
15 object 15.

16 The transmission queue 346 processes the digital object 15 and formulates logical
17 messages, all of which are preferably small, such as about 8 Kbytes. These logical messages
18 are placed onto an output queue, or multiple output queues, and that digital object serves as
19 input to the packetizer module 348.

20 The responsibility of the packetizer module 348 is to take those logical messages,
21 whatever size they are, and make them fit into IP multicast packets in such a way that there
22 is no wasted space. In other words, if there are several small logical messages, they would
23 get packed end-to-end inside of the IP multicast packets.

24 The maximum packet size is preferably selected to stay within the maximum data
25 payload of an Internet frame, currently about 1500 bytes. But more importantly, the length
26 of the data in each of the packets is preferably optimized to be approximately 1442 bytes.
That number is selected in one embodiment to work with the subsequent transmission of that

1 packet over a transmission medium such as a satellite system which incorporates DVB
2 packets and the multi-protocol encapsulation or MPE encodings. The size is chosen to be
3 optimal for data efficiency, so that there is no wasted bandwidth on the satellite signal. Thus,
4 there are no fragmentary DVB (Digital Video Broadcast) packets, in one embodiment.

5 Once the packetizer module 348 has assembled the logical messages and filled up a
6 packet, it forwards that packet to the satellite uplink facility 16, which in this embodiment
7 comprises an IP encapsulator. The IP encapsulator receives packets packetized with IP
8 encapsulation and translates them into bits and pieces that match the DVB packet sizes,
9 which are much smaller.

10 In fact, each DVB packet may hold only approximately 188 bytes of data. The total
11 size of the packet being 204. Some of the packets may be occupied by header information
12 and error correction information. The IP encapsulator outputs those DVB packets in a
13 transmittable state. The packets then pass through standardized hardware for radio frequency
14 conversions and the like, as is commercially available and well known.

15 The packets are then sent from the transmission dish up to geo-synchronous orbit, to
16 the satellite, where they are redirected back down to be received by each of the subscribing
17 local proxy servers 22. Preferably, a satellite receiver at each site receives and recombines
18 the DVB packets into the original IP multicast packets that were fed into the IP encapsulator
19 at the uplink facility.

20 The IP multicast is a standard protocol known in the industry. The packet structure
21 of IP multicast is generally the same as that of IP packets used with TCP/IP or UDP/IP
22 transport protocols. A primary difference, however, is the addressing of where the packet
23 going. That is, the target addressing uses special range IP addresses which indicate that the
24 packets can be received by many receiving stations at the same time.

25 Under this embodiment, a one to many transmission originates at the central proxy
26 server and is delivered to the local proxy servers 22. The satellite receiver or other
transmission medium receiver reconstructs and outputs the original IP multicast packet onto

1 a local Ethernet network, which is preferably a private network, and which may be a single
2 cable between the satellite receiver and the user net in the local proxy cache machine. The
3 local caching module 17 receives a payload from those packets which is processed as
4 illustrated in Figure 10.

5 Referring now to Figure 9, shown therein is a block diagram illustrating one
6 embodiment of the configuration of a series of packets constructed in the transmission of a
7 digital object 15. Initially shown is the transformation of the web objects 15 that are received
8 out of the transmission queue 346 of Figure 8 and inserted into individual object messages
9 360.

10 Each digital object 15 is preferably broken up into multiple object messages 360
11 ranging from 1 to N. Each object message 360 consists of an object message header 362 and
12 a payload 364. Each object message 360 is preferably broken up into a satellite packet 366
13 by the packetizer module 348. The Satellite packets, as previously stated, are preferably
14 1442 bytes in size, but may be any suitable size selected to provide optimal satellite
15 throughput.

16 The satellite packet similarly consists of a packet header 368 and a payload 370.
17 There are 1 to M packets for every given object message. The satellite packets are converted
18 into IP multicast format via industry standard operations. The IP multicast format is then
19 converted into a digital video broadcast (DVB) stream, which is once again an industry
20 standard.

21 The DVB packets can be constructed using a hardware and software solution. An
22 example of this is a DVB packetizer product which is provided by Skystream International
23 of Sunnyvale, California. On the receive side, the DVB packets are received by a satellite
24 modem, which then rebuilds those packets into IP multicast format. The thusly formatted
25 packets are then received by the local proxy server 22, reassembled, and eventually the entire
26 web object 15 is reconstructed using the inverse of the process previously described.

Example of Priority Calculation By the Central Proxy Server

Figure 11 illustrates one method 450 in which the central proxy server 12 of Figure 8 may utilize feedback from local proxy servers 22 to make a global priority calculation for digital objects 15 stored therein. In one embodiment, the central proxy server 12 is configured as described for Figure 8. The method 450 begins at a step 452 and progresses to a step 454 where the central proxy server 12 receives feedback data from the local proxy servers 22. The feedback data is received periodically over a network, typically the Internet 20. Various forms of local feedback that may be transmitted are shown in steps 470 - 477 and will be discussed below.

As indicated at a step 456, the feedback data may indicate a new digital object 15 which is not currently in the priority queue 342. If such an object 15 is indicated by the feedback data as shown by a step 458, the object is placed in the priority queue 342. In the depicted embodiment, this comprises assigning the object 15 a global identification number as indicated at a step 460, requesting the content rating and other meta-data for the object 15 from the communicating data extractor servers 35 as indicated at a step 462. The object 15 is then added to the priority queue 464, in order of its initial priority. This initial priority is preferably a default value. The priority values within the priority queue 342 may be numbers, for instance, between 0 to 255, or more preferably with a higher resolution, such as -32000 to 32000. A value of 10, for instance, may be assigned a new object 15. As discussed, those objects remain in the priority queue until they drop to a sufficiently low value as to be deleted, or reach a sufficiently high value to be passed to the transmission queue. Once an object has been transmitted, it may be placed back in the priority queue 342, but its priority may be decremented to reflect its recent transmission.

Returning to step 454, the feedback data may be received at various times, and may include miss data received as indicated at a step 470, hit data received as indicated at step 472, "top 100" data received as indicated at a step 474, and notification of new versions of transmitted objects received as indicated by a step 476. The hit data may include hit data for

1 objects requested locally by users of the local proxy servers 22, but not yet transmitted by the
2 central proxy server 12, as will be discussed below. In addition, data from external sources
3 such as the data extractor servers 35 may be received as indicated by a step 477. The external
4 sources may be commercially available sites, and may also be servers configured to crawl the
5 web, locating meta-data for objects 15, as well as determining popularity or other information
6 about the digital objects 15.

7 In one simple embodiment, such data is immediately used to recalculate priority of
8 the objects 15 in the priority queue, and the objects 15 are then reprioritized. Nevertheless,
9 in order to avoid content churning, as described in the captioned examples below, such
10 updates may occur only periodically, either at intervals of time, or upon changes of priority
11 of one or more objects of a selected incremental amount. Thus, in the depicted embodiment,
12 the method 450 checks at a step 478 to determine whether it is time to reprioritize yet. That
13 is, has the selected amount of time lapsed, or has the priority of the selected objects changes
14 sufficiently? If not, the feedback data is stored in temporary files to be used in recalculating
15 object global priority once it becomes time to reprioritize. In one embodiment, however,
16 some feedback data is of such high priority, that it is recalculated immediately. Such
17 feedback comprises, in one example, notification from a local cache advisor that an object
18 transmitted from the central proxy server is out of date and that there is a newer version.

19 Such feedback may be gained, for instance, when a local caching module 17 attempts
20 to inject an object into the caching database 24, and the object is rejected because the cache
21 management software 29 (of Figure 8) rejects the object because it already has or is otherwise
22 aware of a newer version and informs the caching module 17 of this fact using the API
23 interface 405. The local proxy server 22 then transmits notification of the newer version to
24 the master proxy server 12, preferably immediately and over the Internet 20.

25 Thus, as indicated at a step 466, such high priority objects are immediately made the
26 subject of a reprioritization, or in a further embodiment, are immediately moved into the
transmission queue 342 for uplink to the satellite 18. In addition, a priority boost may be

1 given to the object as indicated at a step 467. In one embodiment the priority boost is
2 sufficient to achieve immediate injection of the object into the transmission queue 342 is
3 given to the high priority objects.

4 If, on the other hand, it is determined at the step 478 that it is time to reprioritize, the
5 method 450 progresses to a step 480. At the step 480, the recently transmitted feedback data,
6 as well as any such data stored previously in temporary files as discussed is compiled and the
7 priority of all affected objects 15 is recalculated. The objects 15 are then reprioritized in the
8 queue. Typically, the objects are stored in the master cache database 14, and only a short
9 identifier such as a tag is stored in the priority queue, as will be discussed below. The tag is
10 linked by pointers to other data regarding the object 15 in the master cache database 14. It
11 is that identifier tag that is manipulated in the priority queue 342 to represent the newly
12 calculated global priority.

13 After the objects are reprioritized, other house keeping items may also be conducted.
14 As indicated in Figure 11, one such item is to check the satellite transmission rate and to
15 periodically adjust the uplink rate to the satellite accordingly. The new uplink rate may be
16 used to adjust the threshold at which objects 15 are moved from the priority queue 342 into
17 the transmission queue 346, as indicated at a step 484.

18 As indicated at a step 486, the highest priority items, typically those meeting the
19 threshold level set at the step 484, are periodically transmitted to the transmission queue for
20 transmission to the satellite 18. Those objects are uplinked to the satellite as indicated at a
21 step 488. Transmitted with the objects are the global identifier for each object 15, as well
22 as the global popularity for the object 15, and optionally, any meta-data for the object, as
23 indicated by a step 490. Control is then returned to the step 454, and the method 450
24 continually loops in this manner until the central proxy server 12 is shut down.

25 26 Example of Local Proxy Server Operation

1 Referring now to Figure 10, shown therein is a block diagram illustrating the
2 operation of one embodiment of a local caching module 17 operating within a local proxy
3 server 22 of Figure 1. Within Figure 10 is shown a satellite receiver 372. Emanating from
4 the satellite receiver 372 are transmission signals 374 and 376 indicating network
5 transmissions using multicast packets for digital objects 15. The digital objects 15 are shown
6 being received by a receive assembler module 378.

7 Each depicted module represents a software subsystem or component of the local
8 caching module 17. A packet resequencer 378 receives the IP multicast packets as they are
9 transmitted and resenquences the packets in the proper order if needed. The properly
10 sequenced packets are then passed to the message reassembler 379 which extracts the
11 contents of the packets and assembles them into messages and/or digital objects 15.

12 The digital objects 15 are then transmitted via a message dispatcher 380, which is
13 basically a switching mechanism, to an injector module 392. The injector module 392
14 accumulates the object messages, one or more object messages per digital object,
15 reassembles those messages into the web object, and injects that digital object 15 into the
16 local cache database management module 29.

17 The injector module 392 preferably does not wait until all of the object messages for
18 a digital object 15 have been received. As long as they are coming in order and it has the
19 very first message, it proceeds to create that object 15 as far as the cache database
20 management module is concerned and will continue to append each additional object
21 message as it arrives, until the last object message has arrived. If anything is out of order,
22 it waits until it receives the missing portions.

23 A satellite health manager 386 queries the satellite receiver 372 about every fifteen
24 seconds sending SNMP (Simple Network Management Protocol) queries by using the
25 UDP/IP transport and the satellite receiver, if all is well, responds affirmatively.

26 If that is not the case, satellite health manager 386 sends a message via the message
dispatcher 380 to a central proxy server session manager 382. The session manager

1 communicates with the central proxy server 12 over a TCP/IP connection 384. That
2 connection is preferably maintained over the standard Internet and the report of satellite
3 receiver misbehavior can thus reach the central caching module 13 and trigger an alarm or
4 notification such that operations personnel can try to identify the problem and resolve it.

5 The satellite health manager 386 is configured to simultaneously log an error in the
6 systems error log on the local caching module 17. An attention module 402 is also
7 preferably provided. Within the attention module 402 is shown a GUI information module
8 404, an error log status module 406, and a statistics module 408. The attention subsystem
9 402 is responsible for bringing attention of events to human users by displaying information
10 through the graphical user interface to the customer or to any remote administrator that is
11 hooked into the cache database management module and is accessing the specific
12 management panels. The attention module 402 also logs those errors in a local file.

13 A remote console 388 is shown provided for diagnostic purposes. The present
14 invention allows a remote connection over the regular Internet to the local caching module
15 17 with a simple utility which implements what appears to be just a simple command line
16 such as a DOS command line. The system may enter commands of its own through the local
17 caching module 17 that give back responses providing information about internal statistics
18 and performance and other diagnostic information. Those commands may also be used to
19 set or change internal parameters or characteristics of the local caching module 17 for
20 purposes of optimizing its performance.

21 An init/exit module 390 is also shown and is preferably configured to coordinated
22 initialization the various subsystems of Figure 10 with the local cache management module
23 29 when the system is brought on-line. The init/exit module 390 also provides such
24 coordination when the system is taken off-line.

25 As an example of a manner of use of the local caching module 17, a user with a client
26 browser 25 may be browsing the Internet through a local cache database management
module. If the user requests a web object 15 which has not been previously cached by the

1 local cache database management module, the local cache database management module
2 calls up the miss manager subsystem 400, identifies the URL (universal resource locator) of
3 that web object on the Internet, and gives the local caching module 17 an opportunity to take
4 steps to find and gather that digital object and deliver it to the local cache database
5 management module 29 by means of the present system in the manner which has been
6 described up to this point, where the digital object is received over the satellite by the local
7 caching module 17 from the Central proxy server and injected or discarded back into the
8 local cache database management module. Thus, the local cache database management
9 module is given first right of refusal and tries to satisfy the end user request for a particular
10 web option.

11 Example of Hit and Miss Rate Reporting

12 Referring to Figure 10, the local cache database management module 29 is preferably
13 configured to report cache misses to the miss manager 400 of the local caching module 17.
14 In response, the miss manager 400 preferably transmits a message immediately to the central
15 proxy server 12 via the message dispatcher 380 and the central proxy server session manager
16 382 to instruct the central proxy server 12 that the local proxy server 22 does not contain a
17 copy of the web object 15 the end user is seeking.

18 The central proxy server 12 preferably uses the miss information in calculating global
19 miss rate information. That is, the central proxy server 12 uses the miss reports that it
20 receives from the local proxy servers 22 in the overall system 10 to compute the relative
21 priority of web objects 15 in the priority queue 342.

22 Referring back to the local caching module 17 of Figure 10, also provided therein in
23 the depicted embodiment is a hit manager 396. The hit manager 396 is similar to the miss
24 manager 400 in that when a client requests a web object 15 from a local cache database
25 management module 29 and the cache database management module 29 has that object
26 already stored in its cache, a hit is registered. The local proxy server 17 accumulates hit

1 counts on an object-by-object basis as they are reported. It then reports that information at
2 least every few seconds to the central proxy server 12 via the message dispatcher 380 and the
3 central proxy server session manager 382.

4 The local caching module 17 preferably logs hit reports under several different
5 circumstances. Initially, it may only have room to track hits on a certain number of objects
6 at once, which in turn determines the size of the report messages that sends to the central
7 proxy server 12. Once a message content becomes full, it sends the message, allowing it to
8 empty or zero out all of those accumulating buffers, counters, and accumulators.

9 Each web object that the system 10 delivers is preferably assigned a globally unique
10 identifier, and the hit count report utilizes that identifier to be very efficient in forwarding
11 hits to the central proxy server 12. In so doing, it may simply send out an array of structures.
12 In one embodiment, each structure is an array comprised of two elements. One element is
13 the global identification number which represents the web object that the system 10 is
14 tracking. The other element is a count of the number of hits that has just been recorded with
15 a very recent task.

16 As that array of structures fills up or is used up by recording hits for different objects,
17 the current message is sent as soon as all of the array structures are in use or, if not all of
18 them are in use, the partial messages are preferably transmitted no greater than about five
19 seconds apart to make sure that information stays timely until it gets back to the central proxy
20 server 12. That array of information or partial array is also transmitted immediately if one
21 or more of the objects which are being hit has experienced a very large number of hits in a
22 very short period of time. A hit frequency measurement is preferably observed, as well as
23 just accumulations of numbers of total hits.

24 All of these messages that are sent between the central caching module 13 and the
25 local caching module 17, whether they are sent over the traditional Internet using TCP/IP
26 connections or whether they are delivered via multicast transmission, share one common

1 protocol structure or proprietary protocol means. That is, at the beginning of the message,
2 is a header that identifies the overall length of the message and the target of the message.

3 For example, when a miss manager of a local caching module 17 needs to send a miss
4 report, it prepares a message, and a header is appended that will identifies the target as e.g.,
5 "central proxy server subsystem miss handler." When the central proxy server 12 transmits
6 object messages, the header of those messages targets the local caching module injector
7 subsystem and it is the responsibility of the message dispatcher box to switch those messages
8 back and forth between the appropriate subsystems on the local or master.

9 The present invention thus reports hits that have been registered globally on objects
10 that are being tracked by the system 10 and attempts to report those hit counts as quickly and
11 efficiently as possible while they have relevance. When the hit report reaches the central
12 proxy server 12, the hit counts are matched up with the appropriate web object that is being
13 tracked using the global identification number. The hit counts are then added into the prior
14 known hit counts for that object and a new priority for that object can be calculated using
15 miscounts and hit counts to date, as well as optionally, other selected priority and popularity
16 information.

17 Every time a hit or miss report is received by the central proxy server 12 , it
18 accumulates that information and updates the priority totals. Nevertheless, it may not
19 actually transmit, or complete the recalculation of the priority, because doing so requires
20 removing the object from the priority queue and reinserting them at a new position. Given
21 that there will likely be hundreds of thousands of web objects represented from the priority
22 queue at any given time, doing so becomes an expensive proposition in terms of computer
23 processing time. Rather, the potential change in priority is evaluated, and when it crosses a
24 certain threshold or every so many times a report has been received, the object is reinserted
25 into its appropriate position.

26 Relative priorities of objects 15 are preferably transmitted under two circumstances.
First, when the object itself is transmitted, the global priority for that object is transmitted

with it. That information is provided to the local caching module 17 and to the local cache database management module. The local cache database management module at that point preferably incorporates the global popularity data that has been provided into its own internal priority calculations to determine whether it wants to keep the object its local cache on a long-term basis.

The second time that an object's priority is typically transmitted is when the global priority for the object has changed substantially that is, beyond a selected threshold. Since the object data itself has not changed, there is no need to resend the object data, but it is useful to send the very brief, very small administrator message saying the object as identified by its global identification number and has a new global priority for local consideration.

In addition to recording global hit counts on a moment to moment basis for various objects, the hit manager preferably also keeps the longer term totals of those hits. In fact, it preferably keeps a total for a time period of, for example, fifteen minutes. Every object that has had any hits on it during that time period is then evaluated relative to all the other objects in the priority queue. The objects are sorted out and the record of the ordered top 100, 200 or some other selected number of objects are logged to a file in the local cache box and that file is made available for subsequent retrieval by the central caching module 13 and a historical database agent 398.

The historical database engine 398 periodically queries each of the local caching modules 17, to receive all of the accumulated 15 minute logs and data regarding what was most popular in the local caches during that 15 minute time period. This information is used to update all the different local caching modules 17 and also presents a number of opportunities for data mining. For example, the collective "top 100" lists of the associated local proxy servers 22 may be combined into a highly accurate list of what is popular on the Internet and commercially marketed or used in research.

One advantage of the system 10 of the present invention is that there preferably exists a closely integrated relationship between the local caching module 17 and the cache database

Another advantage is that hit counts are accumulated within hit reports at the local cache. Many hit counts may be accumulated before transmission due to time constraints. Thus, a large number of hits on a selected object may be transmitted in a single report.

Example of Tightly Integrated Caching Software

Referring to Figure 10, a number of arrows 405 are shown directed into and out of a cache database management module 29. These arrows in essence represent the functional interface for an application programming interface (API). In one embodiment, the tight integration between the cache database management module 29 (one example of which is an ICS program, as discussed) and the local caching module 17 is conducted through the use of an API. The term “API” as understood in the industry is a collection of programming interfaces that allow one set of software to access services or information directly from another set of software in a standardized manner which separates and protects each side from the other, through an established method of communication. Typically, the two software modules operate on a common server, and their communications are transmitted directly between each other.

The API can be considered to be a type of dynamic binding between two applications operating on a common server or other processor. The terms “integral communication,” “integrally communicate” and “tight interface” represent some sort of binding between two separate applications on a common processor. As disclosed, the binding is preferably a dynamic binding, one example of which is conducted through an API.

An API defines all of the interface information that must be implemented by a vendor in a product. The use of an API-type interface in one embodiment allows the present

1 invention to benefit from close, substantially instantaneous communication between the local
2 caching module 17 and the cache database management module 29. For example, referring
3 to the injector module 392 of Figure 10, arrows pass from the injector module 393 into and
4 out of the cache database management module 29. In one embodiment, four functions are
5 provided which the injector may call on which occur within the cache database management
6 module 29. Each of those functions is called with a short command, and if necessary,
7 accompanying parameters. Those commands are responded to by the performance of the
8 requested function, which may, for instance, include the depositing of data in a selected
9 buffer or memory location.

10 As an additional example, when the injector module 392 receives web objects or
11 other files 15 and needs to transmit those objects to the cache database management module
12 29, it uses commands from selected API instruction sets and passes those commands to the
13 cache database management module 29 to achieve that desired function.

14 Other API functions may include an instruction or command which requests the cache
15 database management module 29 to locate an object 15 within the local cache database 24.
16 Another API instruction may notify the cache database management module 29 that a
17 particular digital object 15 is being sent and is to be saved together with another stored digital
18 object 15. This digital object may not actually be part of the original digital object 15, but
19 may be related information, such as a component of a web page 15a.

20 In one embodiment, the cache database management module 29 may respond to the
21 instruction asynchronously, that is, at varying times. The responses may be immediate, or
22 may be delayed, to acknowledge the results of an initial request, for instance. Other
23 functions may comprise cache database management module one-way communication to the
24 local caching module 17 to provide notification of certain events such as a hit report or a
25 miss report.

26 Thus, under the invention, an API interface is very highly integrated between the local
caching module 17 and the cache database management module 29. This provides the

1 advantages of being able to communicate internally with the cache database management
2 module 29 and being able to receive unique types of information back from the cache
3 database management module, including, for instance, statistics such as hit reports, almost
4 instantaneously.

5 Integration does require some additional support and alteration by the vendor and
6 produces the results of greater flexibility of cache management and information extraction.
7 Additionally, speed is increased, so that reported information is more current. In prior art
8 arrangements, where an software applications residing with different external servers
9 communicate, there are necessarily slowdowns resulting from network transmission delays,
10 processing of network messages in and out of the boxes, and from hardware bottlenecks.
11 These include the use of drivers of the hardware and slow funneling of information that must
12 work its way up through the various hardware protocols and then back down again. These
13 slowdowns are avoided through the tight integration of the present invention. Under the
14 invention, communication to the cache database management module 29 may be as simple
15 as making and responding to a function call.

16 Prior art caching systems typically receive new objects in the order the objects are
17 sent to that caching system. The caching system then just throws out the least recently used
18 digital object to make way for the new digital object coming in a very unintelligent manner.
19 Prior art systems may provide limited feedback on misses, due to the need to request a digital
20 object that is not present in the cache. In such systems, there is no sense of global priority.
21 That is, each remote caching system is unaware of what is popular elsewhere, other than a
22 possibly a rudimentary calculation based upon misses. The prior art caching box responds
23 to transmitted digital objects by making room for this new arrival, and uses it's own best
24 judgement in doing so.

25 Thus, one embodiment of the present invention is based directly on having an
26 integrated API relationship between the cache advisor and the cache software. This allows
the system 10 to have access in real time to prepare information about what is going on inside

1 the cache database management module 29, and also what is going on relative to particular
2 web objects 15 that are being tracked. Given that information, the local caching module 17
3 and the central proxy server 12 collaborate. Each of the local caching modules 17, that are
4 online and subscribed to the system 10 report their popularity information to the central
5 caching module 13, which compiles that information and sends it out together with the rated
6 extracted digital objects 15. The local caching modules 17 then utilize that global popularity
7 data in determining which digital objects 15 to keep and which to discard, optionally in
8 combination with unique local determination factors.

9 Local proxy servers 22 thus act as the eyes and ears across the world. Due to the
10 ability to communicate closely with the cache database management module, the local proxy
11 servers 22 are able to extract at least hits, and optionally, many other types of information
12 from the local cache databases 24. Examples of information that may be extracted from the
13 local caching module 17 under the present invention include acquisition time -- the time that
14 it takes to receive an object over the Internet; information such as what is being deleted out
15 of the local cache, which may indicate local priorities; time data -- digital objects that are hot
16 in a particular area or time zone; hit time -- the time it took the local cache database
17 management module to retrieve a digital object from the Internet; and other particularized
18 information about what digital objects are being kept or thrown out of the local caches. Even
19 the time zone of where that local cache is can be related and associated with the global
20 popularity data.

21 22 Example of Localized Relevance Determination

23 Figure 12 illustrates one embodiment of a method 500 for calculating the local
24 relevance of objects 15 transmitted from the central proxy server 12 and for determining
25 whether to attempt to inject transmitted objects 15 into the local cache database 24. The
26 local relevance determination of Figure 12 allows each local proxy server 22 to make an
individualized assessment of transmitted objects 15 for use in deciding whether to store or

1 continue to store the objects in the attendant local caches 24. The local relevance
2 determination method 500 may be conducted by the local proxy server 22 illustrated and
3 described with respect to Figure 10, and particularly, by the local caching module 17 of the
4 local proxy server 22.

5 The method 500 starts at a step 502 and progresses to a step 506 where digital objects
6 15 are periodically received across the transmission medium from the central proxy server
7 12. Generally, the objects 15 are multicast, and may be transmitted by satellite transmission,
8 as discussed above. Preferably, the server on which the local proxy server 22 operates is
9 capable of multitasking, and consequently, the steps of the method 500 may be happening
10 in parallel, rather than in the illustrated sequence, which is given by way of example for
11 discussion purposes only.

12 When a digital object 15 is received, its global identification number assigned by the
13 central proxy server 12 is also received in the illustrated embodiment, as may be the global
14 popularity calculated for the object 15, and any meta-data that may have been collected for
15 the object 15. Preferably, this data is transmitted together with the object 15, but of course,
16 it could also be transmitted separately. In fact, once an object has been transmitted, the
17 central proxy server 12 continues to update its global popularity for use by the local proxy
18 servers 22, and those updated global popularity figures are periodically transmitted in a
19 continuous manner.

20 The object 15 is received into the packet resynchronizer 378 of Figure 10. The object
21 is resequenced and reassembled, as described above. The local cache advisor software 17
22 must also decide whether to attempt to inject the object 15 into the local cache database 24.
23 As indicated by a decision step 510, if the cache is not full, typically because it has only
24 recently been brought online, the object 15 is immediately injected as indicated at a step 512.
25 If the local cache 24 is operating at capacity, and injecting the object 15 would displace other
26 cached objects 15, it must be determined if the object 15 would be sufficiently popular
locally to do so.

1 In making this local relevance determination, the transmitted global popularity rating
2 and all meta-data are preferably considered. Of course, the local caching module 17 could
3 utilize only selected meta-data, or could omit the global popularity rating from the
4 calculation. Such considerations are indicated by way of example by steps 516-522. For
5 example, global popularity may be considered as indicated at a step 516. Additionally, other
6 quantitative data, referred to collectively herein as "meta-data" may also be considered. For
7 instance, the content rating as provided by the servers 35 of Figure 1, or other sources, may
8 be considered at a step 517, the object type, such as HTML, MP3, streaming video, JPEG,
9 GIF, etc., may be considered at a step 518. The size of the object may be considered at a step
10 519, the language that the object is primarily written in may be considered, as indicated by
11 a step 520, and the geography or origination of the object 15 may be considered at a step 521.

12 In addition, as indicated in a step 522, the cost of origination of the object 15 may
13 also be transmitted for consideration in making the local relevance determination. In one
14 embodiment, the cost of acquisition includes the time it took the central proxy server 12 to
15 acquire the object. Size of the object may also be considered, as well as, in one embodiment,
16 how frequently the object is refreshed. For example, a web page available from a low latency
17 server that is small in size would have a low cost of acquisition, while a web page that is
18 large in size and has a high latency would typically have a high cost of acquisition, making
19 caching of the object 15 more desirable.

20 This data is then compared to local feedback information at a step 523. For instance,
21 in the depicted embodiment, at subsequently described steps 534-538, data is gathered for
22 each object 15 discarded from the local cache database 24. Such data may include the meta-
23 data for that object when discarded, including the objects described at the steps 516-522, and
24 additionally, the time of pendency of the object in the cache may also be considered. This
25 data is analyzed and quantified, and used to determine the likelihood of an object 15
26 remaining in the cache. Local hits and misses on objects of different types may also be
compiled, as discussed above.

1 This actual relevance calculation is made at a step 524. In one embodiment, by way
2 of example, the calculation of step 524 involves the use of analytical methods such as fuzzy
3 logic, including statistical methods or other logical comparative or analytical techniques. For
4 example, various qualities of an object as indicated by its global popularity and meta-data are
5 each considered as a separate dimension. Each dimension is quantified against the history
6 of treatment (e.g., time of pendency of objects of that type before being discarded, proportion
7 of hits on objects of that type, and/or percentage of objects of that type being quickly
8 discarded) of similar objects with that quality, and each dimension is given a rating according
9 to how popular it is to local users of the cache 24. The comparison may be quantitative as
10 well as qualitative, comparing numerical values such as hits and misses and global
11 popularity, as well as taking into account characteristics and local interest in the language of
12 the object, origin of the object, and the like.

13 The various dimensions are then each considered, possibly using analytical tools, in
14 a simple example, by adding the dimensions together, with optional weighting between the
15 various dimensions. Each of the various dimensions, or the cumulative sum of the
16 dimensions may be compared to a threshold value or values. The comparison to the
17 threshold value(s) is indicated at a decision step 526. If the relevance calculation of the
18 object 15 does not meet or exceed the threshold value(s), the object 15 is not injected, and
19 control is returned back to step 504.

20 If, however, the threshold value is met, the object is injected or attempted to be
21 injected into the local cache 24 at a step 528. The injection may be merely attempted,
22 because the local cache 24 may operate under a cache management system 29 that uses its
23 own relevance determination algorithm. If the object does not meet the requirements of the
24 local cache, it may not be accepted. Additionally, the local cache may have various filters
25 therein which may cause the object 15 to be rejected from being cached therein. The cache
26 management system 29 typically displaces one or more objects 15 upon receiving the new
object 15, as has been discussed. The cache management system 29 may use a commercial

1 LRU algorithm for so doing, or may employ a relevancy algorithm similar to that discussed,
2 and may similarly employ the transmitted meta-data or other information compiled by the
3 local caching module 17.

4 If the object 15 is injected, requests for the object are still reported, but are reported
5 as hits, rather than misses. Reporting hits is indicated at a step 530. The hits may be
6 reported together with misses, and in the same manner described above. Hits are preferably
7 collected and compiled locally, and are also transmitted to the central proxy server 12, as
8 discussed. Hits for objects transmitted from the central proxy server 12 are preferably
9 compiled and transmitted, as are in one embodiment, hits on objects filled locally and not
10 transmitted from the central proxy server 12. These hit reports may also be used by the
11 central proxy server 12 in making a global popularity rating for an object 12 that has not yet
12 been transmitted.

13 In the continued operation of the local proxy cache 22, the aforementioned local
14 information is gathered and stored as indicated at a step 532. This may include compiling
15 statistics about discarded objects as illustrated in steps 534-538. For instance, the global
16 popularity of discarded objects may be gathered and compiled at a step 534, the nature of
17 discarded objects may be compiled, at a step 536, and the pendency in the cache of the
18 discarded objects and/or other meta-data measurements may be gathered at a step 538. This
19 data may then be cross-correlated, for instance, to determine the average pendency of objects
20 of a certain language, content type, subject matter, and/or global popularity. This information
21 is then used in making the analysis of step 524 and the decision of step 526.

22 As indicated at a step 540, the threshold value used at the step 526 may be
23 periodically updated. In one example, if objects are being injected too quickly and objects
24 having a higher local demand are being displaced by those objects, the threshold value is
25 increased. If on the other hand, discarded objects have a significantly lower local priority
26 than objects being injected, the threshold value is decreased. The method 502 continually

1 loops, preferably with parallel processing of the various steps, until shut down of the local
2 proxy server 22, at which time it terminates.

4 Example of Transmission of Web Pages in Their Entirety

5 The system 10 of the present invention allows a web page 15a or other digital object
6 15 to be transmitted in its entirety, rather than having to transmit a listing of the contents of
7 the web page 15a and then going back and separately requesting those contents as is
8 conducted in prior art Internet accesses. One feature preferably provided within the cache
9 database management module 29 of the present invention is the ability to examine an HTML
10 web object (such as a web page) 15 and determine its component parts. The HTML object
11 may be a container that describes the complete layout of a particular page 15a that will be
12 seen on the browser window 25 of a user station 30. The container or other such directories
13 typically contain most of the readable text and contain specific instructions, using the
14 hypertext mark up language syntax, to identify other web objects, images, Java applets, or
15 other types of web object that are part of the page.

16 Upon receiving a user request for a digital object 15, the cache database management
17 module 29 preferably looks at the container for the objects 15. The container typically refers
18 to all of the other images and elements that comprise the total Web page experience. The
19 cache database management module 29, upon receiving the HTML object 15, scans through
20 it immediately, parsing out the particular references to other objects that are components of
21 that page 15a. As the cache database management module 29 identifies each component, it
22 initiates a request for the component. The central proxy server 12 then checks to see if the
23 image is already present in the master cache database 24. If it is not, the system 10 initiates
24 a request over the Internet 20 to retrieve that component and continues to do that for every
25 separate object that was indicated as a component of that HTML container object.

26 This feature provides an additional acceleration to the end user 30. The cache
database management module 29 has hopefully retrieved some, if not all, of the component

1 objects that complete a web page by the time that the end user browser 25 has received the
2 original HTML content. The cache database management module 29 preferably begins to
3 analyze and parse the requested object 15 and goes back to the cache database management
4 module 29 and requests all component objects of the web page 15. The entire contents of a
5 web page 15a are then streamed to the web browser 25, without waiting for the web browser
6 25 to request them individually. Without this read-ahead feature, as it is called, on the cache
7 database management module 29, the browser 25, after receiving the original HTML object,
8 would then be required to request each of the component objects with a separate request, and
9 each of those request would result in a cache request, and subsequently, a request across the
10 Internet. Whereas with the read-ahead feature of the cache database management module,
11 hopefully many of those objects 15 are already in the cache database management module
12 and the client browser 25 receives them automatically or alternatively, receives immediate
13 responses when the browser 25 requests those objects 15.

14 This "read-ahead" feature is preferably also provided within the central caching
15 module 13. Accordingly, when a cache miss request arrives from a local caching module 17,
16 and the central caching module 13 is requested to retrieve this web object 15, that web object
17 will generally be the initial reference to a particular web page 15. Thus, the object 15 is
18 preferably retrieved by the cache database management module of the central proxy server
19 12 and handed off to the central caching module 13 along with some identifying information
20 that the received object is an HTML container that may have certain components. The cache
21 database management module 29 of the central proxy server 12 then preferably automatically
22 pre-fetches or pre-locates those components, whether in its local cache or out on the Internet,
23 and notifies the central caching module 13 as each one of those components are identified
24 and located. This allows the central proxy server to be aware of the entire structure of a web
25 page early on.

26 In one embodiment, the HTML object, together with a list of component objects that
accompany the web page 15a are stored together in the cache database management module

1 29, both of the local proxy servers 22 and the central proxy server 12. The system 10 knows
2 about the digital object, and when it transmits the content of that web page 15, it not only
3 sends the digital object, but it preferably also sends, together therewith, data for all of the
4 component objects that it knows about. When that information arrives, it is passed through
5 to the injector subsystem. Part of those messages will be the container, the information that
6 identifies that particular HTML object and represents the whole page. The container
7 preferably also identifies which of the objects that are being received are component objects
8 of the page 15a.

9 With that information about which objects comprise a web page, the injector
10 subsystem begins to inject all of those web objects into the local cache, and when it has
11 completed putting all of those objects in, it then informs the local cache 24 of that
12 relationship. In certain embodiments, the grouping information may come first, or it may
13 come last, but at least, under the invention, the local caching module 17 has the opportunity
14 to inform the local cache that these objects all go together and form a logical entity called a
15 web page. This information is preferably used by the local cache to optimize its storage of
16 those objects on the hard disk to keep them in close proximity for later retrieval if need be,
17 so that retrieval time is held to a minimum.

20 Example of Operation of the Hybrid Priority Determination Scheme

21 As discussed, the present invention preferably includes a hybrid heuristic scheme for
22 making a priority determination regarding whether to store received digital objects 15. The
23 local caching modules 17 can be likened to eyes and ears that report what they see to the
24 central caching module 13. The central caching module 13, in turn, aggregates that
25 information, calculates the global popularities of objects, and keeps other information for
26 future use in deciding what to send to local proxy servers 22. Thus, the central caching

1 module 13 collects data regarding what to send out and completes the full feedback loop of
2 observers and a coordinator and distributor of information.

3
4 Example of Priority determination and Content Churning Prevention

5 The present invention preferably includes a cache overrun feature, also known as an
6 anti-churning algorithm. As an example of the need for this feature, if a local caching
7 module 17 simply forces large quantities of web objects upon the cache database
8 management module 29, the cache database management module could be forced to discard
9 older, less referenced, objects. Given a sufficiently high rate of injecting objects, eventually
10 fill the local cache becomes filled with information that may or may not be of dramatic
11 interest to that local user base and eventually, extracted digital objects 15 begin being pushed
12 out a back end of the process while other, potentially less popular objects 15 are inserted in
13 the front end.

14 Thus, if a cache is occupied merely with injecting objects, it is not performing its
15 primary function, which is to retain content. A balancing point is necessary, and the anti-
16 churning algorithm is thus useful for monitoring the objects 15 that are being discarded from
17 the cache, observing which discarded objects are system-delivered objects and which are not,
18 and observing which objects are present in the cache without being multicast from the central
19 proxy server 12. These objects may have been obtained by the local proxy cache got on its
20 own, which may have provide a miss report which was of too low a global priority to be
21 honored.

22 In other words, for the objects being discarded, we identify which ones are objects
23 that the system 10 did not deliver. We also identify the objects that the system 10 did
24 deliver. From the objects the system 10 did deliver, we identify the global popularity of that
25 object. The relevant amount of system-delivered objects compared to non system-delivered
26 objects may likewise be determined. Using this information, the local proxy server 22 builds
up kind of a feedback loop over time to see how the contents of the cache are affected by

1 injecting objects and by pushing objects out. That is, it examines the priorities associated
2 with system-delivered objects and determines a threshold on a priority basis. Objects below
3 that custom-set threshold are not injected into the local cache database management module
4 while objects above the threshold are ensuring that only lower priority objects are displaced.

5 Web objects are sent by the central caching module 13 on the basis of global
6 popularity, as computed by all of the reports from all the local caching modules 17.
7 Consequently, when a local caching module 17 receives a web object data, attendant to that
8 object is its global popularity rating, which serves as input to the local proxy cache regarding
9 how popular the object might be locally and the local cache can use this information to
10 determine locally how that relates to its own prioritization scheme. So, the feedback loop
11 determines that while objects are inserted into the cache, the cache is predominately
12 discarding everything with a priority of less than, for instance, 2,000 out of 5,000 maximum.
13 This indicates that the higher priority objects were more wanted by that local user base and
14 hence stayed in the local cache. Preferably fuzzy logic is used to establish the threshold.
15 The priority determination can be made quantitatively or qualitatively. This is a way to
16 quantitatively decide what to inject. Significantly, the threshold is set individually and
17 independently by every local proxy server 22.

18 The content churning algorithm of the present invention thus prevents the contents
19 of that local cache from being overrun by an endless stream of system-delivered objects that
20 may be popular elsewhere, but not necessarily popular locally. It allows the local caching
21 module 17 to continually adjust the threshold to consistently result in injecting only those
22 objects 15 that are most likely to be referenced and used by the local user base. The
23 determination is primarily quantitative in that it does not take into account the actual nature
24 of the content of the objects or the quality of those features, the classification of the contents
25 and what is selective about it.

26 When serving a diverse user base, in terms of what kind of contents the users want
to see, particularly when all of those users are going through the same proxy cache, the

1 content churning algorithm may not be as effective. However, it is considered effective in
2 terms of avoiding overrun in terms of the content selection, and it does allow the local cache
3 to avoid being flooded and to continue to store content that is popular locally.

4 The algorithm individualizes the threshold for each local proxy server 22. It has been
5 reported that in some installations of the prior art, for example, cache hit rate, which is the
6 primary measure of efficiency and how much value there is to a cache, actually decreases
7 once the satellite delivered digital objects start being injected. This occurs because the
8 globally popular data has little relevance to the local user base, and so much of it is
9 transmitted, that it pushes out other content that the local cache ordinarily would keep around
10 for its user base.

11 The anti churning algorithm keeps objects from being pushed into the local cache at
12 too fast a rate, and thus prevents wholesale overrun of that local cache. That means that
13 objects that would otherwise stay in the local cache will not be forced out, and in the natural
14 course of events, as determined by that local cache, anything that is injected, if not used by
15 the local user base, it will very quickly be discarded.

16 The anti-churning algorithm of the present invention can be likened to treating the
17 information over the satellite as water from a fire hydrant. For any particular local cache,
18 rather than be overwhelmed by the flow, we simply put a straw in it and tap just the right
19 amount of flow of information.

20 21 Example of Content Localization

22 Content localization under the present invention involves utilizing a feedback
23 mechanism to heuristically construct a model of the nature of the content that the users of the
24 local cache like or do not like. This requires a system for rating the content. Currently,
25 various commercially available products do this.

26 These products can integrate with a proxy caches and provide a database that is
continually updated on a daily basis. Such products typically provide a sizable database, and

1 categorize and rate websites therein in many different categories. The ratings go beyond
2 identifying pornographic and obscene web sites, and also identify other content such as
3 business sites, educational sites, and so forth.

4 Preferably, the system 10 establishes a rating for content that is tracked within the
5 system 10. Consequently, everything stored in the central caching module 13 and the central
6 proxy server 12, preferably has an associated rating. This is preferably accomplished by
7 incorporating one of the described content rating products and integrating that product into
8 the system 10.

9 Once provided with that rating information, the central proxy server 12 sends the
10 content rating with the web object or web page over the satellite 18 and the information is
11 then received by the local caching module 17, which then uses that information in making
12 its own decision whether or not to actually inject that digital object 15 into the local cache
13 24. The local caching module 17 preferably makes that determination based on analyzing
14 the recent history of the types of digital objects that have been discarded by the local cache.
15 It preferably also looks at the content rating of digital objects that have been discarded and
16 from that determines which areas of content are of the least interest to that user base.

17 A local proxy server 22 may also be categorized by its primary user base type. For
18 instance, it may be labeled educational, governmental, etc., and may be programmed to make
19 a correlation between the classifications of its type of users at the individual local caching
20 modules 17 and the classification of subject matter of incoming digital objects 15.

21 Thus, the present invention provides a feedback loop of information regarding the
22 nature of the content being discarded by the local cache. It is significant that every local
23 cache starts out empty. Once it fills up with data, it stays full, and never empties. The
24 purpose of the cache is to hold the most relevant digital objects for the local user base. So,
25 the essence of the content localization and even the anti-churning algorithm involves a
26 feedback loop that monitors what is discarded. That is, the local cache determines what is
of highest value to its user base, and determines statistically what kinds of objects both in

1 nature of their content and in their relative priority around the world are most likely to be in
2 demand by that user base. The priority calculation is local. Objects that do end up being
3 injected into the local cache which turn out not to be relevant to the local user base will very
4 quickly lose whatever initial priority status they had and will be discarded by the local cache
5 as other new objects come in.

6 7 Example of Local Private Intranet

8 Individual corporations or companies may under the invention buy a given amount
9 of band width to transmit their corporate Intranet data. The present invention provides
10 several different methods of doing so. One method currently being utilized is to purchase
11 band width at a specific time to download the corporate data.

12 An alternative, provided under the present invention, is a demand driven system.
13 Working within our current prioritization system, we may maintain priorities for a given
14 Intranet and make sure that the information downloaded is the most pertinent at the given
15 time within that Intranet. Such a system preferably operates on a separate PID (program ID)
16 within the satellite that may be scrambled and secured for that corporate data.

17 Each PID is different logical channel. The system preferably utilizes total bandwidth
18 across several different PID's at once. Additionally, a single PID doesn't actually reserve a
19 specific amount of bandwidth, but rather is simply a data stream identifier. A corporation
20 with an Intranet may have many web based applications for their employees to use in all of
21 their outer offices, wherever they are, and may take advantage of the same demand driven
22 architecture to have those digital objects prioritized and transmitted as needed to be cached
23 at the local office sites.

24 25 Example of Clustering Mechanism

26 One problem encountered is building a central proxy server 12 that is large enough
to be highly efficient. If built with too large of a hard drive, it becomes unreasonably

1 expensive, and if not large enough, it becomes hard to gather all the digital objects that are
2 needed.

3 The solution under the present invention is to create another machine that would be
4 close by the central proxy server 12 and that acts as a local cache. When the central proxy
5 server 12 decides that it needs to delete something for space reasons out of the central proxy
6 server 12, it then sends that digital object 15 using the ICP protocol to this attendant cache
7 server and uses the acquisition time, the time that it took to fetch that object off the Internet,
8 as the priority of that digital object.

9 The attendant local cache then tends to keep around things that were expensive to get
10 so that next time the central proxy server 12 goes looking for an object, those that were
11 expensive to obtain over the Internet are available immediately, once again using the ICP
12 protocol. The present invention optionally provides multiple local proxy servers 22, each
13 with a different range of priorities or acquisition time, for a scalable cluster of caches.

14 The present invention places a local caching module 17 on each of the clustered
15 machines. This allows for transmitting the lower priority objects locally using the same
16 format used over the satellite. But in this case it is just targeted directly across a high speed
17 network to the machines local to the central proxy server 12 machine. This allows the system
18 10 to send the priority of the objects as well. In a nutshell, the cost of acquisition may now
19 be taken into account in the priority scheme of the present invention.

20 When the system 10 pushes the digital object into the secondary cache which has the
21 local caching module 17 operating in it, a global popularity is provided, and the cache
22 database management module uses that global popularity information to help determine how
23 important it is to keep that object around. In essence, the popularity is not being substituted
24 per se. Indeed, the popularity of the object has clearly fallen down a bit, or it wouldn't have
25 been at the bottom of the priority queue in the central proxy server 12 itself. As a result, in
26 this embodiment, the acquisition time, how long it might take to get this object, is substituted
for the global popularity, additionally, size of the object may be considered. If an object is

1 very large, even over a fast network, it may take a significant amount of time to reload the
2 object if it is ever needed. For a small object that is in a very remote part of Chile, for
3 example, which still takes several seconds to get the object would be stored and so that way
4 the things that are most difficult to replace are retained.

6 Example of Global Identifier

7 Regarding representing a digital object by a unique identifier, the key to this is that
8 the identifier is much shorter than the URL or any other identifying information of the object
9 thereby making it much easier to maintain a database with the identifiers and their
10 corresponding popularity data. The identifier is assigned by the central proxy server for
11 every object that is stored therein and/or transmitted to the local caching modules 17.

12 Furthermore the idea of a very short global identifier that is global within the context
13 of the system 10, is considered unique. Being only a few bytes long, it facilitates very
14 efficient reporting from all the local proxy servers 22 when they report object hit counts as
15 described earlier. This enables the present invention to be very efficient in tracking and
16 identifying statistics regarding a particular web object 15, whether on the central proxy server
17 12 or on local caching module 17. Each local proxy servers 22 reference the same unique
18 global identification number when they receive and report information about a particular web
19 object.

21 Example of Racing Return of Data

22 When the local cache records an object miss in the local caching module 17, as
23 indicated before, the cache advisor sends a message to the central caching module 13 to the
24 effect of "this local cache doesn't have this object and the users want it." The central caching
25 module 13 takes that under advisement, so to speak, and decides what it will with it. It may
26 decide to transmit or even re-transmit that object over the satellite depending on the relative
priority of that object to all of the other millions of objects it is working with.

1 It is possible that the response to the user's request may come back over the satellite,
2 and hopefully does so as often as possible. However, if it takes too long, or does not return,
3 the local cache will, itself, also be requesting the information over the standard Internet, all
4 the way to the remote server that has the data. It may receive that digital object back, before
5 any information comes over the satellite. So there are two possible request paths: (1) through
6 the system 10 from the local caching module 17 to central caching module 13 to possible
7 multicast of the digital object 15 over the satellite, or (2) directly from the local cache itself
8 to the content publisher webserver and back. It can then be likened to a race to see which one
9 comes first.

10 It is important to note that if the local caching module 17 becomes nonfunctional, or
11 the satellite system is disrupted for some reason, the local cache continues to function, albeit
12 at probably a lower efficiency because its going to have to use up more bandwidth upstream
13 as it talks to the Internet. Thus, this redundancy is also a safety factor. The present invention
14 is thus a self-healing system. Moreover, the transmission of requested data is seamless to the
15 user. Due both the race concept and the concept of redundant system, if the satellite becomes
16 inoperative for whatever reason, the user still receives uninterrupted service.

17 18 Example of Redundant Transmission Paths

19 A related aspect to the redundant communication paths for delivering digital objects
20 back to the local cache is that in the event of failure of the satellite communication path, the
21 central caching module 13 can elect to send the digital objects it normally would have
22 transmitted by satellite over the terrestrial Internet. It may have to throttle back on the
23 amount of digital objects that it sends, due to the lack of capability to multicast a single
24 packet of data which is received by many local proxy servers 22 at once. Instead, it sends out
25 one copy of a packet to every local cache, which is a bandwidth consumer. Therefore, in one
26 embodiment, the central proxy server 12 is limited to sending only the most important, most
globally popular items when transmitted over terrestrial lines. However, doing so still

1 achieve some measure of operation to assist the local proxy servers 22 until the alternative
2 delivery mechanism, such as a satellite or a multicast backbone comes back on line.

4 Example of Slip-Through Transmission Queues

5 Referring again to Figure 8, shown therein is the box labeled transmission queue 346.
6 The transmission queue 346 manages logical messages that represent all of the web objects
7 that have been chosen for transmission. Those objects have been chosen because they have
8 the highest relative priority of all of the objects in the central proxy server 12, and once they
9 have been chosen, they are placed on the transmission queue 346 in the order in which they
10 were chosen. The transmission queue 346 subsystem then preferably starts to gather the
11 actual component parts of the objects. Up until this point the central proxy server 12 has just
12 been manipulating a small data structure, a pointer or tag, that defines each of the objects.
13 The actual digital object is cached by the local cache database management module, but the
14 priority of the object is manipulated using the small data structure that has associated
15 therewith all the relevant information about that object. It is those small data structures that
16 are put onto the transmission queue 346.

17 When it becomes time to actually access the real data of an object so it can be
18 transmitted, the software of the transmission queue 346 starts with the first object on the
19 queue. The software requests through the API to the cache database management module
20 to "open this object or access and read data from this object". It goes through this cycle and
21 over time reads as much as it can until it reaches the end of the data for that object.

22 The process of opening and reading data for an object sometimes is not instantaneous.
23 Sometimes the object data is out on the hard drive of the system, rather than in RAM, and
24 there is usually a latency, of possibly several milliseconds. So if the first object on the
25 transmission queue 346 has initiated these read operations, but there is still no data, then the
26 transmission queue 346 software moves on to the second object and goes through the same

1 process, opening the object for access, reading the data, etc., but now the difference might
2 be that all of that information is in memory and hence the access to it is instantaneous.

3 What preferably happens under the present invention is that the second object may
4 be transmitted before the first. Or at least as much of it as is available in memory at that
5 time. It will in effect "slip through" the queue up to the front, ahead of the one that
6 technically had a greater priority, simply because the system is not yet able to get the data for
7 that object of higher priority.

8 The transmission queue 346 preferably continues working down the line of those
9 digital objects and sending digital objects on a basis of what not only has the greatest original
10 priority, but also on a basis of what is available in its entirety. Consequently, at any point
11 during the scan along the queue, certain objects are available, and when there is no further
12 data available for objects of higher priority, then the objects of highest priority that are
13 available get processed into the logical messages, or object messages and are then placed on
14 an output queue. The output queue is then read by the packetizer in the subsystem
15 transmitter across the satellite. The system 10 does its best to send the highest priority digital
16 objects as quickly as it can but it does not wait and waste time if a high priority object is not
17 quite available yet. It takes the next best thing.

18 19 Example of Heuristics Consideration of Time Since Transmission

20 Another unique feature of the system 10 is that the priority calculation in one
21 embodiment takes into account the time since the transmission of an object. Potential items
22 that could be calculated from the cache database management module at the local caching
23 module 17 have been previously discussed. Correspondent to that concept is the priority
24 calculation of the present invention taking into account different informational categories.
25 These categories include those recited previously that can be referenced uniquely from the
26 cache database management module, and specifically, the time since transmission.
Additionally, acquisition time, subject matter, and other available information may be taken

1 into account when making a priority determination at the central proxy server 12.
2 Conversely, when making a priority determination at the local cache, the quantitative
3 calculation of the present invention preferably comes into play.

4 5 Examples of a Binary Tree

6 A further aspect of the present invention is a methodology for maintaining priority
7 of digital objects 15. In one embodiment this methodology comprise the use of a binary tree
8 and a bubbling mechanism. Alternatives include the use of hashing, linear queues, etc., but
9 are not believed to be as efficient as the binary tree and bubbling mechanism.

10 The system 10 preferably maintains the aforementioned priority queue 342 of all
11 objects 15 that are in the central proxy server 12. It is preferable to keep these objects sorted
12 in order of global popularity. Global popularity may also be referred to herein as "priority."
13 It should be noted that global popularity and transmission priority may be become slightly
14 separate values at some point. Nonetheless, the queue maintains the objects in need for a
15 priority of transmission or a global popularity index, which in one embodiment are the same.

16 Regarding the binary tree approach, the sorting key for inserting objects into the tree
17 is the value that results from the priority calculation, combined with a small element of
18 randomization. If the system simply relied upon a priority number that had a fairly small
19 degree of precision, say 6 digit numbers or 4 digit numbers, the possibility of coming up with
20 different objects that all have the same transmission priority would be very high. For
21 example, every object that came in on its first miss would all get the same priority.

22 The sorting key used under the present invention for each object that is inserted into
23 the binary tree is the result of the priority calculation. However, the least significant digits
24 of that value are replaced with as random number, and the overall priority value is a 32 bit
25 floating point value. This value is transformed into a 32 bit number, and the lower several
26 bits, possibly 8 to 10 bits, are put in a random number, so in place of the lower 10 bits is
inserted a random number between 0 and 1,023.

1 By randomizing object ratings in this manner, the system still maintains the
2 significant part of the priority, the significant portion, reflecting hits and misses on the object.
3 Yet, when objects with what would otherwise be the same value are present, those objects
4 are nonetheless distinguished ever so slightly by the randomization. The randomization is
5 an efficiency that allows the tree to stay balanced, to have a minimal depth. This means that
6 to search for any object, in order for example, to insert or delete the object, the system only
7 has to process at most down to the maximum depth of the tree. If all objects that all have the
8 same priority value were inserted, they would create the equivalent of a rope ladder that
9 would just keep going and going and going. The maximum depth of the tree would get
10 extremely large and at that point, the system would simply be running a linear list, which we
11 have noted is a poor choice for this type of mechanism.

12 So, by using the randomization, the present invention has very successfully caused
13 the objects to be sorted into the tree and allows the tree to remain fairly well pruned. That
14 is, the tree does not have little branches that go running off. This adds great efficiency to the
15 lookup and the manipulation of objects of the priority queue.

16 The invention can alternatively employ a balanced binary tree algorithm, such as a
17 red/black tree algorithm, which incorporates additional processing every time you insert or
18 delete an object to detect whether or not surrounding nodes in the tree on certain branches
19 need to be reshuffled in order to avoid the long linear chain. This is considered to be overly
20 complicated, however, for achieving the desired results. Simply randomizing the keys
21 produces that effect automatically on a statistical basis to as much a degree as necessary.
22 This allows the system to use a simple binary tree algorithm which has the fastest possible
23 manipulation time for inserting and deleting objects.

24 Example of Bubbling Mechanism

25 Conceptually, using the bubbling mechanism, a large number of objects bubble
26 around in the priority queue in competition to see which gets the highest priority with all of

1 the different hit and miss counts reporting in real time from all of the local proxy servers 22
2 and affecting the priority ratings of the objects. The objects with the highest priority are
3 transmitted first.

4 5 Example of Self-Adjusting Uplink Rate to Speed of Satellite

6 In some circumstances, not all requested objects can be transmitted. The amount
7 transmitted is simply a function of the transmission bandwidth available. The present
8 invention has been designed to feed the output to that packetizer module 348 of figure 8,
9 which needs to receive the objects just at the right speed to keep the satellite channel full of
10 data. So working backwards in the transmission queue 346 to the priority queue, objects are
11 taken off the top of the priority queue only as fast as they are needed to keep the outgoing
12 bandwidth from having empty spots, which would be wasteful. The present invention is self-
13 adjusting to the speed of the satellite transmission. That is, the rate of providing objects to
14 the transmission queue 346 is adjusted automatically according to the speed of the satellite
15 transmission.

16 17 Example of Selectively Timing the Calculation of Priority

18 Even though the present invention implements a very efficient algorithm for
19 managing the binary tree that makes up the line structure of the priority queue, a large
20 number of hit and miss reports are generated under the invention, especially hit reports.
21 Once an object has been multicast and delivered everywhere, the system 10 receives mostly
22 hit reports and it is considered advantageous to cut down on the number of times that the
23 system moves an object around on the priority queue.

24 Every time hits and misses of a particular object are recorded, the calculated priority
25 for that object changes. In one embodiment, each time the priority of an object changes, the
26 object is moved in the priority queue. This involves deleting it from the binary tree and then
reinserting it using the new priority as the key value. In a further embodiment, an object is

1 relocated in the queue only every N times that the priority of an object is recalculated. The
2 object may also be relocated if the newly calculated priority has increased a selected amount
3 over what it was the last time the object was moved. This may be based on a percentage
4 increase, or on how quickly the object has increased in priority since the last time it was
5 moved, which would indicate an accelerating interest in this object. Under this embodiment,
6 a balance is struck between moving an object when necessary and not short-changing any
7 object, because it is not being moved every single time a hit is reported. It is preferable not
8 to short-change an object, allowing the objects to move up in priority as rapidly as they
9 deserve, while still saving processor time.

11 Example of Pre-Delivery of Objects Based Upon Historical Analysis

12 A further aspect of the present invention is the pre-delivery of objects based on
13 historical analysis. Pre-delivering objects is considered to be a very important part of the
14 whole system, because doing so allows digital objects to be transmitted to local proxy servers
15 22 before, in many cases, the user base of a local proxy server 22 asks for them. That effect
16 is multiplied and facilitated by having a large number of local proxy servers 22.

17 The central proxy server 12 and optionally a nearby database machine preferably keep
18 a historical record of the popularity of different web objects, pages, or web sites to a selected
19 degree of resolution. A record is kept of the most popular sites (e.g., the "top 100") for each
20 individual local cache for every 15 minute period during the day. The system 10 analyzes
21 that data and determines over all on a more global or system-wide basis when and where the
22 demand for particular objects is greatest.

23 In essence, the system 10 examines the daily cycles of usage patterns and determines,
24 for example, that at 6:00 am New York Time, there was a large upswing in demand for Wall
25 Street Journal pages. That demand tapers off in three hours and picks back up in the evening
26 perhaps, slightly, and then it tapers off significantly, almost to nothing during the night time
hours, relative to New York. The system in one embodiment analyzes such information,

1 collecting the data for at least one 24 hour period, and then continues to modify it and smooth
2 it over time, using it, deliver time sensitive objects at or before their demand peaks.
3 Continuing the example, Wall Street Journal Pages are delivered just prior to the time they
4 are in high demand. Again, this is a general principal. Preferably, the system 10 pre-delivers
5 the most popular pages just before the times when they have high or peak demands
6 approaching. Doing so provides a significant savings of bandwidth of all the local proxy
7 servers 22 and increases the end user experience, because the new browser responds
8 immediately with the requested web object.

10 Example of Popularity Ratings and Historical Mining of Data

11 The central proxy server 12 is in one embodiment provided with a software agent that
12 is configured to do analysis. The software agent could even be on a separate machine, if
13 desired. Preferably, a location is provided to store the data and sufficient processing power
14 is provided when needed to analyze it. A communication channel for informing the central
15 proxy server 12 of the objects that it should start working into the priority queue at a certain
16 time is similarly provided. That same database of information can also be mined
17 independently in order to report information back to content publishers or others. For
18 instance, reports can be provided on various patterns of usage of digital objects 15 and so
19 forth. The system 10 can thus provide top 100 ratings, daily usage patterns, things that would
20 help network system engineers plan their network layouts and bandwidth capacities, and the
21 like. Such information may help content publishers redesign their content or distribute it in
22 different ways, such as on different sites to avoid a crunch on any one particular site.

24 Example of Transmission of Low Priority Data

25 Further novel aspects of the invention include placing a notation on digital objects
26 15 transmitted over the satellite which are of low priority. For instance, user stations 30 may
request a digital object 15, that does not have a sufficient global popularity to be retained,

1 even by the requesting local proxy server 22. Accordingly, a mechanism is provided for the
2 requesting stations to identify low priority web objects that are transmitted in response to
3 their specific requests.

4 One manner of doing this under the invention is to group a number of local caching
5 modules 17 that have requested a web site, whether that be one or more and when the
6 queuing reaches a priority to where there are sufficient of those to transmit the web page, to
7 then place a tag at the first of the packets, designating the local caching modules 17 that
8 specifically requested the web site. Problematically, however, a further requesting web site
9 may then request the same information right after it was sent over the satellite. The
10 requesting local caching module 17 is not on the exclusive send list. Thus, much like ships
11 passing in the night, the late requester will have received that information or will be receiving
12 it the same time it is requesting it but does not know that it is receiving it.

13 A solution to this is an alternative manner in which the list of requesting stations is
14 replaced with an infinitely scalable mechanism. The mechanism is much easier to use than
15 keeping a list of the requesting local caching modules 17. This mechanism in one
16 embodiment comprises sending to the designating location in the packet a hashed copy of the
17 URL. The local caching modules 17 correspondingly each keep a list, a revolving list of the
18 URL's requested, preferably also hashed. This list revolves in that once it is full, every new
19 URL that comes in pushes an old URL off the bottom. Accordingly, whenever a new web
20 object comes across the satellite, it is accompanied by a global identification number,
21 popularity information including optionally, meta-data, and the hashed URL.

22 Of course, the hashed URL could be substituted with the local caching module 17
23 identification numbers, but the hashed URL is considered to be more effective. The local
24 caching module 17 then compares the hashed URLs on its list against the hashed URL
25 coming across the satellite dish. In this manner the local caching module 17 does not need
26 to unhash the URL as it comes and can make the comparison quickly. In addition, this
eliminates the situation where the local caching module 17 sends out a request and

1 subsequently receives an object which it had previously requested but for which the local
2 caching module's ID number is not included thereon.

3 Use of the hashing algorithm, by it's nature, poses a slight chance of conflicts.
4 Nevertheless, it is believed that at least a 99% success rate of correlating the hashed URL
5 with the URL on the local caching module list will result from the present invention. Thus,
6 transmitting the hashed URL significantly assists in the acceleration of content of the present
7 invention, as the local caching module 17 need not necessarily continually go back to its hit
8 and miss routine to find a requested web object.

10 Example of Operation of Redundant Central Proxy Servers

11 Under the present invention, it is preferred that the central proxy server 12 be
12 provided with a backup in case of failure. In one embodiment, depicted in Figure 13, the
13 Internet content delivery system 10 of Figure 1 has been modified to include a plurality of
14 servers 12 each configured redundantly to operate as a central proxy server 12. One server
15 12 operates as a master 46, while a second server 12, preferably configured substantially in
16 the same manner as the first, operates as a backup 48. While two redundant servers 12 are
17 shown, it should be appreciated that any number of redundant servers 12 may be utilized
18 under the present invention.

19 The plurality of redundant servers 12 may be co-located and may communicate
20 locally. Nevertheless, for additional safety purposes, it is preferred that the two servers 12
21 be located geographically remote from each other. So locating the servers 46, 48 reduces the
22 possibility that an event, such as a power outing, fire, earthquake, or the like, that might
23 render one server inoperable or unable to communicate with the local proxy servers 22 would
24 also prevent the second server from operating normally. Thus, the servers 12 may be in a
25 separate city, state, or even country from each other. It is preferred that the geographically
26 separated servers 12 communicate over a private communications channel 47, such as a
leased T1 line, but, of course, the servers 12 could also communicate over other types of

1 communication mediums, such as the Internet 20. The servers 12 are preferably loosely
2 coupled, maintaining communications over the communications channel 47. Through that
3 communications, the servers 12 preferably synchronize with each other.

4 One of the servers 12 is preferably in operation as the master 46, while the second is
5 in operation as the backup 48 at all times. The master 46 preferably operates in the manner
6 described herein for the central proxy server 12, while the backup server 48 awaits a failure
7 of the master 46, and upon such failure, assumes the role of the master 46. One server may
8 be dedicated as the master 46, or the plurality of redundant servers 12 may be alternately
9 operated as the master 46.

10 One problem that the inventors have overcome in operating a redundant server system
11 in conjunction with the Interned content acceleration system 10 of the present invention is
12 that the plurality of servers could conceivably lose contact with each other and could thus
13 lose track of which is operating as the master 46 and which is operating as the backup 48.
14 For instance, the two servers could go down, and come up but fail to establish
15 communications with each other. In such a case, the two servers might both attempt to act
16 as the master 46, confusing the local proxy servers 22. Additionally, a server 12 acting as
17 the master 46 could go off-line, and come back on but fail for some reason to communicate
18 with the other server 12 which is now operating as the master 46. Both could thus believe
19 itself to be the master and attempt to communicate as such with the local proxy servers 22.

20 To avoid such an occurrence, a redundancy algorithm 49 is preferably provided and
21 operates within each of the redundant servers 12. The redundancy algorithm 49 arbitrates
22 which of the servers is to be the master 46 at any time and which is to be the backup 48. In
23 further embodiments, the redundancy algorithm also provides a methodology for assuring
24 that only one of the servers 46, 48 operates as the master at any one time.

25 One embodiment of a unique method 550 for maintaining a single master and a single
26 backup at any one time is illustrated in Figure 14. The method 550 of Figure 14 starts at a
step 552 and progresses to a step 554, where the server 12 on which the method 550 is

1 operating receives a token. The token may be stored within the server or may be provided
2 to the server 12 manually or from another redundant server 12. The token is preferably used
3 to indicate whether the server 12 is to operate as the master 46 or the backup 48 and for use
4 in notifying the local proxy servers 22 which of the redundant servers 12 is the master 46.

5 At a step 556, the two servers 12 communicate with each other to arbitrate which is
6 to operate currently as the master and which is to operate as the backup. Because both
7 servers 46, 48 preferably communicate with each to maintain substantially the same content,
8 this decision may be arbitrary. It may also be based upon which of the servers has had the
9 majority of up-time, preserving the longevity of the servers, or which has the greatest
10 capacity. For instance, if one server has less storage space than the other, it may be generally
11 operated in backup mode.

12 This decision is made at a decision step 558, where the method 550 branches. If the
13 server 46, 48 is to operate as the master, the method 550 branches to a step 560. At the step
14 560, the token is preferably synchronized between the two servers 12. Thus, the servers 46,
15 48 may at this stage contain identical tokens. At a step 562, the token is transmitted from the
16 server 12 to the local proxy servers 22. The local proxy servers 22 are thus informed as to
17 which of the servers 46, 48 is the master. The server 12 then operates normally, as indicated
18 at a step 564. This operation is, in one embodiment, as described above with respect to
19 Figures 1 - 12.

20 The server 12 operates normally until a failure occurs as indicated at a step 566.
21 When a failure occurs, the backup server 48 will typically lose communications with the
22 master 46 and will assume the role of master 46. Accordingly, the server 12 maintains its
23 state upon such a failure, typically with CMOS memory, and restarts at a step 568 when the
24 event that caused the off-line condition to occur is rectified. The server 12 then establishes
25 communication with the other server(s) 12, one of which has been acting as the master 46
26 in its absence, and arbitrates which of the servers 12 is to continue as the master 46 at a step
570. The method 550 then returns to the step 558.

Returning to the step 558, if the decision therein is that the server 12 is to be the backup 48, then the method 550 progresses to a step 571. At the step 571, the tokens are synchronized between the two servers 12. As indicated at a step 572, the server 12 continually receives updates from the master 46 over the communication medium 47. Thus, the two (or more) servers 12 are maintained with substantially identical contents. The backup 48 also continually monitors communications with the master 46, and if those communications cease for a selected amount of time, for example, 20 seconds, the backup 48 will deem that the master 46 has failed and gone off-line.

The server 12 then increments the token as indicated at a step 576. The token is then transmitted to the local proxy caches at a step 578, informing the local proxy caches 22 that the server 12 is now the master 46. If the local proxy caches receive two versions of the token, because, for instance, both servers 12 are operating, but cannot establish communication with each other to arbitrate which is to be the master, the server 12 with the token that has the highest value is deemed by the local proxy caches 22 to be the master 48.

The server 12 then changes mode to operation as the master 48 as indicated at a step 580 and operates as the master server 46 until communications can be established with the other server 12 as indicated at a step 582. When this happens, the servers 12 once again arbitrate which of the servers 12 will be the master as indicated at a step 584, and the method 550 returns to the step 558. The method 550 operates upon each redundant server 12, and continually loops until the server 12 is shut down by an operator, at which point the method 550 terminates.

The present invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.